THE ART OF COMPUTER PROGRAMMING

VOLUME 4 PRE-FASCICLE 0A

A DRAFT OF SECTION 7: INTRODUCTION TO COMBINATORIAL SEARCHING

DONALD E. KNUTH *Stanford University*

ADDISON–WESLEY

Internet
Stanford GraphBase
MMIX

Internet page http://www-cs-faculty.stanford.edu/~knuth/taocp.html contains current information about this book and related books.

See also http://www-cs-faculty.stanford.edu/~knuth/sgb.html for information about *The Stanford GraphBase*, including downloadable software for dealing with the graphs used in many of the examples in Chapter 7.

See also http://www-cs-faculty.stanford.edu/~knuth/mmixware.html for downloadable software to simulate the MMIX computer.

Zeroth printing (revision 3), 30 January 2008

FOLLAND
PASCAL
Internet

PREFACE

To put all the good stuff into one book is patently impossible,
and attempting even to be reasonably comprehensive
about certain aspects of the subject is likely to lead to runaway growth.
— GERALD B. FOLLAND, "Editor's Corner" (2005)

La dernière chose qu'on trouve en faisant un ouvrage
est de savoir celle qu'il faut mettre la première.
— BLAISE PASCAL, *Pensées* 740 (c. 1660)

THIS BOOKLET contains draft material that I'm circulating to experts in the field, in hopes that they can help remove its most egregious errors before too many other people see it. I am also, however, posting it on the Internet for courageous and/or random readers who don't mind the risk of reading a few pages that have not yet reached a very mature state. *Beware:* This material has not yet been proofread as thoroughly as the manuscripts of Volumes 1, 2, and 3 were at the time of their first printings. And those carefully-checked volumes, alas, were subsequently found to contain thousands of mistakes.

Given this caveat, I hope that my errors this time will not be so numerous and/or obtrusive that you will be discouraged from reading the material carefully. I did try to make the text both interesting and authoritative, as far as it goes. But the field is vast; I cannot hope to have surrounded it enough to corral it completely. So I beg you to let me know about any deficiencies that you discover.

To put the material in context, this pre-fascicle contains the opening remarks intended to launch a long, long chapter on combinatorial algorithms. Chapter 7 is planned to be by far the longest single chapter of *The Art of Computer Programming*; it will eventually fill at least three volumes (namely Volumes 4A, 4B, and 4C), assuming that I'm able to remain healthy. Like the second-longest chapter (Chapter 5), it begins with pump-priming introductory material that comes before the main text, including dozens of exercises to get the ball rolling. A long voyage lies ahead, and some important provisions need to be brought on board before we embark. Furthermore I want to minimize the shock of transition between Chapter 6 and the new chapter, because Chapter 6 was originally written and published more than thirty years ago.

Chapter 7 proper, which follows the material in the present pre-fascicle, begins with Section 7.1: Zeros and Ones. Section 7.1 is another sort of introduction, at a different level; it has four subsections about Boolean and bitwise computations, appearing respectively in pre-fascicles 0b, 0c, 1a, and 1b. The next part, 7.2, is about generating all possibilities, and it begins with Section

7.2.1: Generating Basic Combinatorial Patterns. Fascicles for Section 7.2.1 have already appeared in print. Section 7.2.2 will deal with backtracking in general. And so it will go on, if all goes well; an outline of the entire Chapter 7 as currently envisaged appears on the `taocp` webpage that is cited on page ii.

This introductory section has turned out to have more than twice as many exercises as I had originally planned. But many of them are quite simple, intended to reinforce the reader's understanding of basic definitions, or to acquaint readers with the joys of *The Stanford GraphBase*. Other exercises were simply irresistible, as they cried out to be included here — although, believe it or not, I did reject more potential leads than I actually followed up.

My notes on combinatorial algorithms have been accumulating for more than forty years, so I fear that in several respects my knowledge is woefully behind the times. Please look, for example, at the exercises that I've classed as research problems (rated with difficulty level 46 or higher), namely exercises 15, 16, 67, and 125; I've also implicitly mentioned or posed additional unsolved questions in the answers to exercises 7 and 133(m). Are those problems still open? Please inform me if you know of a solution to any of these intriguing questions. And of course if no solution is known today but you do make progress on any of them in the future, I hope you'll let me know.

I urgently need your help also with respect to some exercises that I made up as I was preparing this material. I certainly don't like to receive credit for things that have already been published by others, and most of these results are quite natural "fruits" that were just waiting to be "plucked." Therefore please tell me if you know who deserves to be credited, with respect to the ideas found in exercises 3, 25, 32, 35, 72, 84, 108, 116, and 135, and/or the answer to exercises 105.

Thanks to Jeff Dean of Google for letting me look at the statistics of five-letter words in the Internet at the beginning of 2004, and to Robin Wilson of the Open University for his careful reading and many detailed suggestions.

I shall happily pay a finder's fee of \$2.56 for each error in this draft when it is first reported to me, whether that error be typographical, technical, or historical. The same reward holds for items that I forgot to put in the index. And valuable suggestions for improvements to the text are worth 32¢ each. (Furthermore, if you find a better solution to an exercise, I'll actually reward you with immortal glory instead of mere money, by publishing your name in the eventual book:–)

Cross references to yet-unwritten material sometimes appear as '00'; this impossible value is a placeholder for the actual numbers to be supplied later.

Happy reading!

Stanford, California D. E. K.
28 April 2007

The author is especially grateful to the Addison–Wesley Publishing Company for its patience in waiting a full decade for this manuscript from the date the contract was signed.
— FRANK HARARY, *Graph Theory* (1968)

Stanford GraphBase
Dean
Google
five-letter words
Internet
Wilson
Knuth
HARARY

Johnson
Chambers
combinatorial explosion
IBM Type 650 computer

Preface to Volume 4 (draft)

The title of Volume 4 is *Combinatorial Algorithms*, and when I proposed it I was strongly inclined to add a subtitle: *The Kind of Programming I Like Best*. My editors have decided to tone down such exuberance, but the fact remains that programs with a combinatorial flavor have always been my favorites.

On the other hand I've often been surprised to find that, in many people's minds, the word "combinatorial" is linked with computational difficulty. Indeed, Samuel Johnson, in his famous dictionary of the English language (1755), said that the corresponding noun "is now generally used in an ill sense." Colleagues tell me tales of woe, in which they report that "the combinatorics of the situation defeated us." Why is it that, for me, combinatorics arouses feelings of pure pleasure, yet for many others it evokes pure panic?

It's true that combinatorial problems are often associated with humongously large numbers. Johnson's dictionary entry also included a quote from Ephraim Chambers, who had stated that the total number of words of length 24 or less, in a 24-letter alphabet, is 1,391,724,288,887,252,999,425,128,493,402,200. The corresponding number for a 10-letter alphabet is 11,111,111,110; and it's only 3905 when the number of letters is 5. Thus a "combinatorial explosion" certainly does occur as the size of the alphabet grows from 5 to 10 to 24 and beyond.

Computing machines have become tremendously more powerful throughout my life. As I write these words, I know that they are being processed by a computer whose speed is more than 100,000 times faster than the IBM Type 650 computer to which I'm dedicating these books, and whose memory capacity is also more than 100,000 times greater. Tomorrow's machines will be even faster and more capacious. But these amazing advances have not diminished people's craving for answers to combinatorial questions; quite the contrary. Our once-unimaginable ability to compute so rapidly has raised our expectations, and whetted our appetite for more — because, in fact, the size of a combinatorial problem can increase more than 100,000-fold when n simply increases by 1.

Combinatorial algorithms can be defined informally as techniques for the high-speed manipulation of combinatorial objects such as permutations or graphs. We typically try to find patterns or arrangements that are the best possible ways to satisfy certain constraints. The number of such problems is vast, and the art of writing such programs is especially important and appealing because a single good idea can save years or even centuries of computer time.

Indeed, the fact that good algorithms for combinatorial problems can have a terrific payoff has led to terrific advances in the state of the art. Many problems that once were thought to be intractable can now be polished off with ease, and many algorithms that once were known to be good have now become better. Starting about 1970, computer scientists began to experience a phenomenon that we called "Floyd's Lemma": Problems that seemed to need n^3 operations could actually be solved in $O(n^2)$; problems that seemed to require n^2 could be handled in $O(n \log n)$; and $n \log n$ was often reducible to $O(n)$. More difficult problems saw a reduction in running time from $O(2^n)$ to $O(1.5^n)$ to $O(1.3^n)$, etc. Other problems remained difficult in general, but they were found to have important special cases that are much simpler. Many combinatorial questions that I once thought would never be answered have now been resolved, and these breakthroughs are due mainly to improvements in algorithms rather than to improvements in processor speeds.

Floyd's Lemma

By 1975, such research was advancing so rapidly that a substantial fraction of the papers published in leading journals of computer science were devoted to combinatorial algorithms. And the advances weren't being made only by people in the core of computer science; significant contributions were coming from workers in electrical engineering, artificial intelligence, operations research, mathematics, physics, statistics, and other fields. I was trying to complete Volume 4 of *The Art of Computer Programming*, but instead I felt like I was sitting on the lid of a boiling kettle: I was confronted with a combinatorial explosion of another kind, a prodigious explosion of new ideas!

This series of books was born at the beginning of 1962, when I naïvely wrote out a list of tentative chapter titles for a 12-chapter book. At that time I decided to include a brief chapter about combinatorial algorithms, just for fun. "Hey look, most people use computers to deal with numbers, but we can also write programs that deal with patterns." In those days it was easy to give a fairly complete description of just about every combinatorial algorithm that was known. And even by 1966, when I'd finished a first draft of about 3000 handwritten pages for that already-overgrown book, fewer than 100 of those pages belonged to Chapter 7. I had absolutely no idea that what I'd foreseen as a sort of "salad course" would eventually turn out to be the main dish.

The great combinatorial fermentation of 1975 has continued to churn, as more and more people have begun to participate. New ideas improve upon the older ones, but rarely replace them or make them obsolete. So of course I've had to abandon any hopes that I once had of being able to surround the field, to write a definitive book that sets everything in order and provides one-stop shopping for everyone who has combinatorial problems to solve. It's almost never possible to discuss a subtopic and say, "Here's the final solution: end of story." Instead, I must restrict myself to explaining the most important principles that seem to underlie all of the efficient combinatorial methods that I've encountered so far. At present I've accumulated more than twice as much raw material for Volume 4 as for all of Volumes 1–3 combined.

This sheer mass of material implies that the once-planned "Volume 4" must actually become several physical volumes. You are now looking at Volume 4A. Volumes 4B and 4C will exist someday, assuming that I'm able to remain healthy; and (who knows?) there may also be Volumes 4D, 4E, ...; but surely not 4Z.

My plan is to go systematically through the files that I've amassed since 1962 and to tell the stories that I believe are still waiting to be told, to the best of my ability. I can't aspire to completeness, but I do want to give proper credit to all of the pioneers who have been responsible for key ideas; so I won't scrimp on historical details. Furthermore, whenever I learn something that I think is likely to remain important 50 years from now, something that can also be explained elegantly in a paragraph or two, I can't bear to leave it out. Conversely, difficult material that requires a lengthy proof is beyond the scope of these books, unless the subject matter is truly fundamental.

OK, it's clear that the field of Combinatorial Algorithms is vast, and I can't cover it all. What are the most important things that I'm leaving out? My biggest blind spot, I think, is geometry, because I've always been much better at visualizing and manipulating algebraic formulas than objects in space. Therefore I don't attempt to deal in these books with combinatorial problems that are related to computational geometry, such as close packing of spheres, or clustering of data points in n-dimensional Euclidean space, or even the Steiner tree problem in the plane. More significantly, I tend to shy away from polyhedral combinatorics, and from approaches that are based primarily on linear programming, integer programming, or semidefinite programming. Those topics are treated well in many other books on the subject, but they rely on geometrical intuition. Purely combinatorial developments are easier for me to understand.

I also must confess a bias against algorithms that are efficient only in an asymptotic sense, algorithms whose superior performance doesn't begin to "kick in" until the size of the problem exceeds the size of the universe. A great many publications nowadays are devoted to algorithms of that kind. I can understand why the contemplation of ultimate limits has intellectual appeal and carries an academic cachet; but in *The Art of Computer Programming* I tend to give short shrift to any methods that I would never consider using myself in an actual program. (There are, of course, exceptions to this rule, especially with respect to basic concepts in the core of the subject. Some impractical methods are simply too beautiful and/or too insightful to be excluded; others provide instructive examples of what *not* to do.)

Furthermore, as in earlier volumes of this series, I'm intentionally concentrating almost entirely on *sequential* algorithms, even though computers are increasingly able to carry out activities in parallel. I'm unable to judge what ideas about parallelism are likely to be useful five or ten years from now, let alone fifty, so I happily leave such questions to others who are wiser than I. Sequential methods, by themselves, already test the limits of my own ability to discern what the artful programmers of tomorrow will want to know.

The main decision that I needed to make when planning how to present this material was whether to organize it by problems or by techniques. Chapter 5 in Volume 3, for example, was devoted to a single problem, the sorting of data into order; more than two dozen techniques were applied to different aspects of that problem. Combinatorial algorithms, by contrast, involve many different problems, which tend to be attacked with a smaller repertoire of techniques. I

spheres
clustering
Steiner tree
polyhedral combinatorics
linear programming
integer programming
semidefinite programming
sequential

finally decided that a mixed strategy would work better than any pure approach. Thus, for example, these books treat the problem of finding shortest paths in Section 7.3, and problems of connectivity in Section 7.4.1; but many other sections are devoted to basic techniques, such as the use of Boolean algebra (Section 7.1), backtracking (Section 7.2), matroid theory (Section 7.6), or dynamic programming (Section 7.7). The famous Traveling Salesrep Problem, and other classic combinatorial tasks related to covering, coloring, and packing, have no sections of their own, but they come up several times in different places as they are treated by different methods.

I've mentioned great progress in the art of combinatorial computing, but I don't mean to imply that all combinatorial problems have actually been tamed. When the running time of a computer program goes ballistic, its programmers shouldn't expect to find a silver bullet for their needs in this book. The methods described here will often work a great deal faster than the first approaches that a programmer tries; but let's face it: Combinatorial problems get huge very quickly. We can even prove rigorously that a certain small, natural problem will *never* have a feasible solution in the real world, although it is solvable in principle (see the theorem of Stockmeyer and Meyer in Section 7.1.2). In other cases we cannot prove as yet that no decent algorithm for a given problem exists, but we know that such methods are unlikely, because any efficient algorithm would yield a good way to solve thousands of other problems that have stumped the world's greatest experts (see the discussion of NP-completeness in Section 7.9).

Experience suggests that new combinatorial algorithms will continue to be invented, for new combinatorial problems and for newly identified variations or special cases of old ones; and that people's appetite for such algorithms will also continue to grow. The art of computer programming continually reaches new heights when programmers are faced with challenges such as these. Yet today's methods are also likely to remain relevant.

Most of this book is self-contained, although there are frequent tie-ins with the topics discussed in Volumes 1–3. Low-level details of machine language programming have been covered extensively in those volumes, so the algorithms in the present book are usually specified only at an abstract level, independent of any machine. However, some aspects of combinatorial programming are heavily dependent on low-level details that didn't arise before; in such cases, all examples in this book are based on the `MMIX` computer, which supersedes the `MIX` machine that was defined in early editions of Volume 1. Details about `MMIX` appear in a paperback supplement to that volume called *The Art of Computer Programming*, Volume 1, Fascicle 1; they're also available on the Internet, together with downloadable assemblers and simulators.

Another downloadable resource, a collection of programs and data called *The Stanford GraphBase*, is cited extensively in the examples of this book. Readers are encouraged to play with it, in order to learn about combinatorial algorithms in what I think will be the most efficient and most enjoyable way.

Incidentally, while writing the introductory material at the beginning of Chapter 7, I was pleased to note that it was natural to mention some work of

shortest paths
connectivity
Traveling Salesrep Problem
Stockmeyer
Meyer
NP-completeness
MMIX
Internet
Stanford GraphBase

my Ph.D. thesis advisor, Marshall Hall, Jr. (1910–1990), as well as some work of *his* thesis advisor, Oystein Ore (1899–1968), as well as some work of *his* thesis advisor, Thoralf Skolem (1887–1963). Skolem's advisor, Axel Thue (1863–1922), was already present in Chapter 6.

I'm immensely grateful to the hundreds of readers who have helped me to ferret out numerous mistakes that I made in early drafts of this volume, which were originally posted on the Internet and subsequently printed in paperback fascicles. But I fear that other errors still lurk among the details collected here, and I want to correct them as soon as possible. Therefore I will cheerfully pay $2.56 to the first finder of each technical, typographical, or historical error. The `taocp` webpage cited on page ii contains a current listing of all corrections that have been reported to me.

Stanford, California D. E. K.
April 2007

Naturally, I am responsible for the remaining errors—
although, in my opinion, my friends could have caught a few more.
— CHRISTOS H. PAPADIMITRIOU, *Computational Complexity* (1995)

A note on references. References to *IEEE Transactions* include a letter code for the type of transactions, in boldface preceding the volume number. For example, '*IEEE Trans.* **C-35**' means the *IEEE Transactions on Computers*, volume 35. The IEEE no longer uses these convenient letter codes, but the codes aren't too hard to decipher: '**EC**' once stood for "Electronic Computers," '**IT**' for "Information Theory," '**SE**' for "Software Engineering," and '**SP**' for "Signal Processing," etc.; '**CAD**' meant "Computer-Aided Design of Integrated Circuits and Systems."

Hall
Ore
Skolem
Thue
Internet
Knuth
PAPADIMITRIOU
IEEE Transactions

Knuth
Bach

Hommage à Bach.

CHAPTER SEVEN

BASSANIO
Shakespeare
HOLMES
Doyle
TARJAN
puzzles
LOYD, JR.
Langford pairings++
combinatorics–

COMBINATORIAL SEARCHING

You shall seeke all day ere you finde them,
& when you have them, they are not worth the search.
— BASSANIO, in *The Merchant of Venice* (Act I, Scene 1, Line 117)

Amid the action and reaction of so dense a swarm of humanity,
every possible combination of events may be expected to take place,
and many a little problem will be presented which may be striking and bizarre.
— SHERLOCK HOLMES, in *The Adventure of the Blue Carbuncle* (1892)

The field of combinatorial algorithms is too vast to cover
in a single paper or even in a single book.
— ROBERT E. TARJAN (1976)

While jostling against all manner of people
it has been impressed upon my mind that the successful ones
are those who have a natural faculty for solving puzzles.
Life is full of puzzles, and we are called upon
to solve such as fate throws our way.
— SAM LOYD, JR. (1927)

COMBINATORICS is the study of the ways in which discrete objects can be arranged into various kinds of patterns. For example, the objects might be $2n$ numbers $\{1, 1, 2, 2, \ldots, n, n\}$, and we might want to place them in a row so that exactly k numbers occur between the two appearances of each digit k. When $n = 3$ there is essentially only one way to arrange such "Langford pairs," namely 231213 (and its left-right reversal); similarly, there's also a unique solution when $n = 4$. Many other types of combinatorial patterns are discussed below.

Five basic types of questions typically arise when combinatorial problems are studied, some more difficult than others.

i) Existence: Are there any arrangements X that conform to the pattern?
ii) Construction: If so, can such an X be found quickly?
iii) Enumeration: How many different arrangements X exist?
iv) Generation: Can all arrangements X_1, X_2, ... be visited systematically?
v) Optimization: What arrangements maximize or minimize $f(X)$, given an objective function f?

Each of these questions turns out to be interesting with respect to Langford pairs.

For example, consider the question of existence. Trial and error quickly reveals that, when $n = 5$, we cannot place $\{1, 1, 2, 2, \ldots, 5, 5\}$ properly into ten positions. The two 1s must both go into even-numbered slots, or both into odd-numbered slots; similarly, the 3s and 5s must choose between two evens or two odds; but the 2s and 4s use one of each. Thus we can't fill exactly five slots of each parity. This reasoning also proves that the problem has no solution when $n = 6$, or in general whenever the number of odd values in $\{1, 2, \ldots, n\}$ is odd.

In other words, Langford pairings can exist only when $n = 4m-1$ or $n = 4m$, for some integer m. Conversely, when n does have this form, Roy O. Davies has found an elegant way to construct a suitable placement (see exercise 1).

How many essentially different pairings, L_n, exist? Lots, when n grows:

$$\begin{aligned} L_3 &= 1; & L_4 &= 1;\\ L_7 &= 26; & L_8 &= 150;\\ L_{11} &= 17{,}792; & L_{12} &= 108{,}144;\\ L_{15} &= 39{,}809{,}640; & L_{16} &= 326{,}721{,}800;\\ L_{19} &= 256{,}814{,}891{,}280; & L_{20} &= 2{,}636{,}337{,}861{,}200;\\ L_{23} &= 3{,}799{,}455{,}942{,}515{,}488; & L_{24} &= 46{,}845{,}158{,}056{,}515{,}936. \end{aligned} \tag{1}$$

[The values of L_{23} and L_{24} were determined by M. Krajecki, C. Jaillet, and A. Bui in 2004 and 2005; see *Studia Informatica Universalis* **4** (2005), 151–190.] A seat-of-the-pants calculation suggests that L_n might be roughly of order $(4n/e^3)^{n+1/2}$ when it is nonzero (see exercise 5); and in fact this prediction turns out to be basically correct in all known cases. But no simple formula is apparent.

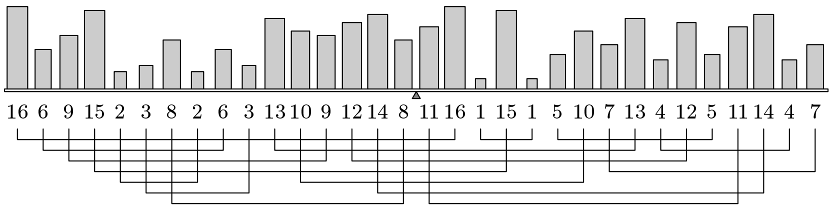
The problem of Langford arrangements is a simple special case of a general class of combinatorial challenges called *exact cover problems.* In Section 7.2.2.1 we shall study an algorithm called "dancing links," which is a convenient way to generate all solutions to such problems. When $n = 16$, for example, that method needs to perform only about 3200 memory accesses for each Langford pair arrangement that it finds. Thus the value of L_{16} can be computed in a reasonable amount of time by simply generating all of the pairings and counting them.

Notice, however, that L_{24} is a *huge* number — roughly 5×10^{16}, or about 1500 MIP-years. (Recall that a "MIP-year" is the number of instructions executed per year by a machine that carries out a million instructions per second, namely 31,556,952,000,000.) Therefore it's clear that the exact value of L_{24} was determined by some technique that did *not* involve generating all of the arrangements. Indeed, there is a much, much faster way to compute L_n, using polynomial algebra. The instructive method described in exercise 6 needs $O(4^n n)$ operations, which may seem inefficient; but it beats the generate-and-count method by a whopping factor of order $\Theta((n/e^3)^{n-1/2})$, and even when $n = 16$ it runs about 20 times faster. On the other hand, the exact value of L_{100} will probably never be known, even as computers become faster and faster.

We can also consider Langford pairings that are *optimum* in various ways. For example, it's possible to arrange sixteen pairs of weights $\{1, 1, 2, 2, \ldots, 16, 16\}$ that satisfy Langford's condition and have the additional property of being "well-

Davies
Krajecki
Jaillet
Bui
exact cover problems
dancing links
mems
MIP-years
well-balanced

balanced," in the sense that they won't tip a balance beam when they are placed in the appropriate order:



(2)

In other words, $15.5\cdot16+14.5\cdot6+\cdots+0.5\cdot8 = 0.5\cdot11+\cdots+14.5\cdot4+15.5\cdot7$; and in this particular example we also have another kind of balance, $16+6+\cdots+8 = 11+16+\cdots+7$, hence also $16\cdot16+15\cdot6+\cdots+1\cdot8 = 1\cdot11+\cdots+15\cdot4+16\cdot7$.

Moreover, the arrangement in (2) has *minimum width* among all Langford pairings of order 16: The connecting lines at the bottom of the diagram show that no more than seven pairs are incomplete at any point, as we read from left to right; and one can show that a width of six is impossible. (See exercise 7.)

What arrangements $a_1a_2\ldots a_{32}$ of $\{1,1,\ldots,16,16\}$ are the *least* balanced, in the sense that $\sum_{k=1}^{32} ka_k$ is maximized? The maximum possible value turns out to be 5268. One such pairing — there are 12,016 of them — is

$$2\ 3\ 4\ 2\ 1\ 3\ 1\ 4\ 16\ 13\ 15\ 5\ 14\ 7\ 9\ 6\ 11\ 5\ 12\ 10\ 8\ 7\ 6\ 13\ 9\ 16\ 15\ 14\ 11\ 8\ 10\ 12. \qquad (3)$$

A more interesting question is to ask for the Langford pairings that are smallest and largest in lexicographic order. The answers for $n = 24$ are

$$\{\texttt{abacbdecfgdoersfpgqtuwxvjklonhmirpsjqkhltiunmwvx},\ \texttt{xvwsquntkigrdapaodgiknqsvxwutmrpohljcfbecbhmfejl}\} \qquad (4)$$

if we use the letters `a`, `b`, …, `w`, `x` instead of the numbers 1, 2, …, 23, 24.

We shall discuss many techniques for combinatorial optimization in later sections of this chapter. Our goal, of course, will be to solve such problems without examining more than a tiny portion of the space of all possible arrangements.

Orthogonal latin squares. Let's look back for a moment at the early days of combinatorics. A posthumous edition of Jacques Ozanam's *Recreations mathematiques et physiques* (Paris: 1725) included an amusing puzzle in volume 4, page 434: "Take all the aces, kings, queens, and jacks from an ordinary deck of playing cards and arrange them in a square so that each row and each column contains all four values and all four suits." Can you do it? Ozanam's solution, shown in Fig. 1 on the next page, does even more: It exhibits the full panoply of values and of suits also on both main diagonals. (Please don't turn the page until you've given this problem a try.)

By 1779 a similar puzzle was making the rounds of St. Petersburg, and it came to the attention of the great mathematician Leonhard Euler. "Thirty-six officers of six different ranks, taken from six different regiments, want to march in a 6×6 formation so that each row and each column will contain one officer of each rank and one of each regiment. How can they do it?" Nobody was able to

width
lexicographic order
Pohl, Ira Sheldon
orthogonal latin squares–
latin squares–
Ozanam
playing cards
Euler

Græco-Latin square

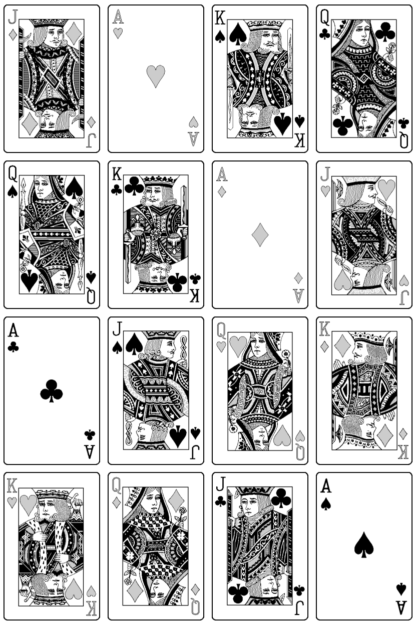


Fig. 1. Disorder in the court cards: No agreement in any line of four. (This configuration is one of many ways to solve a popular eighteenth-century problem.)

find a satisfactory marching order. So Euler decided to resolve the riddle — even though he had become nearly blind in 1771 and was dictating all of his work to assistants. He wrote a major paper on the subject [eventually published in *Verhandelingen uitgegeven door het Zeeuwsch Genootschap der Wetenschappen te Vlissingen* **9** (1782), 85–239], in which he constructed suitable arrangements for the analogous task with n ranks and n regiments when $n = 1$, 3, 4, 5, 7, 8, 9, 11, 12, 13, 15, 16, ...; only the cases with $n \bmod 4 = 2$ eluded him.

There's obviously no solution when $n = 2$. But Euler was stumped when $n = 6$, after having examined a "very considerable number" of square arrangements that didn't work. He showed that any actual solution would lead to many others that look different, and he couldn't believe that all such solutions had escaped his attention. Therefore he said, "I do not hesitate to conclude that one cannot produce a complete square of 36 cells, and that the same impossibility extends to the cases $n = 10$, $n = 14$ ... in general to all oddly even numbers."

Euler named the 36 officers $a\alpha$, $a\beta$, $a\gamma$, $a\delta$, $a\epsilon$, $a\zeta$, $b\alpha$, $b\beta$, $b\gamma$, $b\delta$, $b\epsilon$, $b\zeta$, $c\alpha$, $c\beta$, $c\gamma$, $c\delta$, $c\epsilon$, $c\zeta$, $d\alpha$, $d\beta$, $d\gamma$, $d\delta$, $d\epsilon$, $d\zeta$, $e\alpha$, $e\beta$, $e\gamma$, $e\delta$, $e\epsilon$, $e\zeta$, $f\alpha$, $f\beta$, $f\gamma$, $f\delta$, $f\epsilon$, $f\zeta$, based on their regiments and ranks. He observed that any solution would amount to having two *separate* squares, one for Latin letters and another for Greek. Each of those squares is supposed to have distinct entries in rows and columns; so he began by studying the possible configurations for $\{a, b, c, d, e, f\}$, which he called *Latin squares*. A Latin square can be paired up with a Greek square to form a "Græco-Latin square" only if the squares are *orthogonal* to each other, meaning that no (Latin, Greek) pair of letters can be found together in more than one place when the squares are superimposed. For example, if we let $a = \mathtt{A}$, $b = \mathtt{K}$, $c = \mathtt{Q}$, $d = \mathtt{J}$, $\alpha = \clubsuit$, $\beta = \spadesuit$, $\gamma = \diamondsuit$, and $\delta = \heartsuit$, Fig. 1 is equivalent to the Latin, Greek, and Græco-Latin squares

$$\begin{pmatrix} d & a & b & c \\ c & b & a & d \\ a & d & c & b \\ b & c & d & a \end{pmatrix}, \begin{pmatrix} \gamma & \delta & \beta & \alpha \\ \beta & \alpha & \gamma & \delta \\ \alpha & \beta & \delta & \gamma \\ \delta & \gamma & \alpha & \beta \end{pmatrix}, \text{ and } \begin{pmatrix} d\gamma & a\delta & b\beta & c\alpha \\ c\beta & b\alpha & a\gamma & d\delta \\ a\alpha & d\beta & c\delta & b\gamma \\ b\delta & c\gamma & d\alpha & a\beta \end{pmatrix}. \qquad (5)$$

Of course we can use *any* n distinct symbols in an $n\times n$ Latin square; all that matters is that no symbol occurs twice in any row or twice in any column. So we might as well use numeric values $\{0,1,\ldots,n-1\}$ for the entries. Furthermore we'll just refer to "latin squares" (with a lowercase "l"), instead of categorizing a square as either Latin or Greek, because orthogonality is a symmetric relation.

Euler's assertion that two 6×6 latin squares cannot be orthogonal was verified by Thomas Clausen, who reduced the problem to an examination of 17 fundamentally different cases, according to a letter from H. C. Schumacher to C. F. Gauss dated 10 August 1842. But Clausen did not publish his analysis. The first demonstration to appear in print was by G. Tarry [*Comptes rendus, Association française pour l'avancement des sciences* **29**, part 2 (1901), 170–203], who discovered in his own way that 6×6 latin squares can be classified into 17 different families. (In Section 7.2.3 we shall study how to decompose a problem into combinatorially inequivalent classes of arrangements.)

Euler's conjecture about the remaining cases $n = 10$, $n = 14$, $\ldots$ was "proved" three times, by J. Petersen [*Annuaire des mathématiciens* (Paris: 1902), 413–427], by P. Wernicke [*Jahresbericht der Deutschen Math.-Vereinigung* **19** (1910), 264–267], and by H. F. MacNeish [*Annals of Math.* **23** (1922), 221–227]. Flaws in all three arguments became known, however; and the question was still unsettled when computers became available many years later. One of the very first combinatorial problems to be tackled by machine was therefore the enigma of 10×10 Græco-Latin squares: Do they exist or not?

In 1957, L. J. Paige and C. B. Tompkins programmed the SWAC computer to search for a counterexample to Euler's prediction. They selected one particular 10×10 latin square "almost at random," and their program tried to find another square that would be orthogonal to it. But the results were discouraging, and they decided to shut the machine off after five hours. Already the program had generated enough data for them to predict that at least 4.8×10^{11} hours of computer time would be needed to finish the run!

Shortly afterwards, three mathematicians made a breakthrough that put latin squares onto page one of major world newspapers: R. C. Bose, S. S. Shrikhande, and E. T. Parker found a remarkable series of constructions that yield orthogonal $n\times n$ squares for all $n > 6$ [*Proc. Nat. Acad. Sci.* **45** (1959), 734–737, 859–862; *Canadian J. Math.* **12** (1960), 189–203]. Thus, after resisting attacks for 180 years, Euler's conjecture turned out to be almost entirely wrong.

Their discovery was made without computer help. But Parker worked for UNIVAC, and he soon brought programming skills into the picture by solving the problem of Paige and Tompkins in less than an hour, on a UNIVAC 1206 Military Computer. [See *Proc. Symp. Applied Math.* **10** (1960), 71–83; **15** (1963), 73–81.]

Clausen
Schumacher
Gauss
Tarry
Petersen
Wernicke
MacNeish
Paige
Tompkins
SWAC
Bose
Shrikhande
Parker
UNIVAC 1206 Military Computer

Parker
Paige
Tompkins
SWAC
search tree
mems (memory accesses)
Euler
transversals+

Let's take a closer look at what the earlier programmers did, and how Parker dramatically trumped their approach. Paige and Tompkins began with the following 10×10 square L and its unknown orthogonal mate(s) M:

$$L = \begin{pmatrix} 0&1&2&3&4&5&6&7&8&9\\ 1&8&3&2&5&4&7&6&9&0\\ 2&9&5&6&3&0&8&4&7&1\\ 3&7&0&9&8&6&1&5&2&4\\ 4&6&7&5&2&9&0&8&1&3\\ 5&0&9&4&7&8&3&1&6&2\\ 6&5&4&7&1&3&2&9&0&8\\ 7&4&1&8&0&2&9&3&5&6\\ 8&3&6&0&9&1&5&2&4&7\\ 9&2&8&1&6&7&4&0&3&5 \end{pmatrix} \quad\text{and}\quad M = \begin{pmatrix} 0&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 1&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 2&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 3&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 4&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 5&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 6&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 7&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 8&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup\\ 9&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup&\sqcup \end{pmatrix}. \tag{6}$$

We can assume without loss of generality that the rows of M begin with 0, 1, $\ldots$, 9, as shown. The problem is to fill in the remaining 90 blank entries, and the original SWAC program proceeded from top to bottom, left to right. The top left $\sqcup$ can't be filled with 0, since 0 has already occurred in the top row of M. And it can't be 1 either, because the pair $(1,1)$ already occurs at the left of the next row in (L, M). We can, however, tentatively insert a 2. The digit 1 can be placed next; and pretty soon we find the lexicographically smallest top row that might work for M, namely 0214365897. Similarly, the smallest rows that fit below 0214365897 are 1023456789 and 2108537946; and the smallest legitimate row below them is 3540619278. Now, unfortunately, the going gets tougher: There's no way to complete another row without coming into conflict with a previous choice. So we change 3540619278 to 3540629178 (but that doesn't work either), then to 3540698172, and so on for several more steps, until finally 3546109278 can be followed by 4397028651 before we get stuck again.

In Section 7.2.3, we'll study ways to estimate the behavior of such searches, without actually performing them. Such estimates tell us in this case that the Paige–Tompkins method essentially traverses an implicit search tree that contains about 2.5×10^{18} nodes. Most of those nodes belong to only a few levels of the tree; more than half of them deal with choices on the right half of the sixth row of M, after about 50 of the 90 blanks have been tentatively filled in. A typical node of the search tree probably requires about 75 mems (memory accesses) for processing, to check validity. Therefore the total running time on a modern computer would be roughly the time needed to perform 2×10^{20} mems.

Parker, on the other hand, went back to the method that Euler had originally used to search for orthogonal mates in 1779. First he found all of the so-called *transversals* of L, namely all ways to choose some of its elements so that there's exactly one element in each row, one in each column, and one of each value. For example, one transversal is 0859734216, in Euler's notation, meaning that we choose the 0 in column 0, the 8 in column 1, $\ldots$, the 6 in column 9. Each transversal that includes the k in L's leftmost column represents a legitimate way to place the ten k's into square M. The task of finding transversals is, in fact, rather easy, and the given matrix L turns out to have exactly 808 of them; there are respectively $(79, 96, 76, 87, 70, 84, 83, 75, 95, 63)$ transversals for $k = (0, 1, \ldots, 9)$.

Once the transversals are known, we're left with an exact cover problem of 10 stages, which is much simpler than the original 90-stage problem in (6). All we need to do is cover the square with ten transversals that don't intersect — because every such set of ten is equivalent to a latin square M that is orthogonal to L.

The particular square L in (6) has, in fact, exactly one orthogonal mate:

$$\begin{pmatrix} 0&1&2&3&4&5&6&7&8&9\\ 1&8&3&2&5&4&7&6&9&0\\ 2&9&5&6&3&0&8&4&7&1\\ 3&7&0&9&8&6&1&5&2&4\\ 4&6&7&5&2&9&0&8&1&3\\ 5&0&9&4&7&8&3&1&6&2\\ 6&5&4&7&1&3&2&9&0&8\\ 7&4&1&8&0&2&9&3&5&6\\ 8&3&6&0&9&1&5&2&4&7\\ 9&2&8&1&6&7&4&0&3&5 \end{pmatrix} \perp \begin{pmatrix} 0&2&8&5&9&4&7&3&6&1\\ 1&7&4&9&3&6&5&0&2&8\\ 2&5&6&4&8&7&0&1&9&3\\ 3&6&9&0&4&5&8&2&1&7\\ 4&8&1&7&5&3&6&9&0&2\\ 5&1&7&8&0&2&9&4&3&6\\ 6&9&0&2&7&1&3&8&4&5\\ 7&3&5&1&2&0&4&6&8&9\\ 8&0&2&3&6&9&1&7&5&4\\ 9&4&3&6&1&8&2&5&7&0 \end{pmatrix}. \qquad (7)$$

The dancing links algorithm finds it, and proves its uniqueness, after doing only about 1.7×10^8 mems of computation, given the 808 transversals. Furthermore, the cost of the transversal-finding phase, about 5 million mems, is negligible by comparison. Thus the original running time of 2×10^{20} mems — which once was regarded as the inevitable cost of solving a problem for which there are 10^{90} ways to fill in the blanks — has been reduced by a further factor of more than 10^{12}(!).

We will see later that advances have also been made in methods for solving 90-level problems like (6). Indeed, (6) turns out to be representable directly as an exact cover problem (see exercise 17), which the dancing links procedure of Section 7.2.2.1 solves after expending only 1.3×10^{11} mems. Even so, the Euler–Parker approach remains about a thousand times better than the Paige–Tompkins approach. By "factoring" the problem into two separate phases, one for transversal-finding and one for transversal-combining, Euler and Parker essentially reduced the computational cost from a product, T_1T_2, to a sum, T_1+T_2.

The moral of this story is clear: Combinatorial problems might confront us with a huge universe of possibilities, yet we shouldn't give up too easily. A single good idea can reduce the amount of computation by many orders of magnitude.

Puzzles versus the real world. Many of the combinatorial problems we shall study in this chapter, like Langford's problem of pairs or Ozanam's problem of the sixteen honor cards, originated as amusing puzzles or "brain twisters." Some readers might be put off by this emphasis on recreational topics, which they regard as a frivolous waste of time. Shouldn't computers really be doing useful work? And shouldn't textbooks about computers be primarily concerned with significant applications to industry and/or world progress?

Well, the author of the textbook you are reading has absolutely no objections to useful work and human progress. But he believes strongly that a book such as this should stress *methods* of problem solving, together with mathematical ideas and *models* that help to solve many different problems, rather than focusing on the reasons why those methods and models might be useful. We shall learn many beautiful and powerful ways to attack combinatorial problems, and the elegance

exact cover problem
dancing links
Euler
Parker
Paige
Tompkins
–combinatorics
puzzles++
Langford
Ozanam
recreation
Knuth

of those methods will be our main motivation for studying them. Combinatorial challenges pop up everywhere, and new ways to apply the techniques discussed in this chapter arise every day. So let's not limit our horizons by attempting to catalog in advance what the ideas are good for.

For example, it turns out that orthogonal latin squares are enormously useful, particularly in the design of experiments. Already in 1788, François Cretté de Palluel used a 4×4 latin square to study what happens when sixteen sheep — four each from four different breeds — were fed four different diets and harvested at four different times. [*Mémoires d'Agriculture* (Paris: Société Royale d'Agriculture, trimestre d'été, 1788), 17–23.] The latin square allowed him to do this with 16 sheep instead of 64; with a Græco-Latin square he could also have varied another parameter by trying, say, four different quantities of food or four different grazing paradigms.

But if we had focused our discussion on his approach to animal husbandry, we might well have gotten bogged down in details about breeding, about root vegetables versus grains and the costs of growing them, etc. Readers who aren't farmers might therefore have decided to skip the whole topic, even though latin square designs apply to a wide range of studies. (Think about testing five kinds of pills, on patients in five stages of some disease, five age brackets, and five weight groups.) Moreover, a concentration on experimental design could lead readers to miss the fact that latin squares also have important applications to coding and cryptography (see exercises 18–24).

Even the topic of Langford pairing, which seems at first to be purely recreational, turns out to have practical importance. T. Skolem used Langford sequences to construct Steiner triple systems, which we have applied to database queries in Section 6.5 [see *Math. Scandinavica* **6** (1958), 273–280]; and in the 1960s, E. J. Groth of Motorola Corporation applied Langford pairs to the design of circuits for multiplication. Furthermore, the algorithms that efficiently find Langford pairs and latin square transversals, such as the method of dancing links, apply to exact cover problems in general; and the problem of exact covering has great relevance to crucial problems such as the equitable apportionment of voter precincts to electoral districts, etc.

The applications are not the most important thing, and neither are the puzzles. Our primary goal is rather to get basic concepts into our brains, like the notions of latin squares and exact covering. Such notions give us the building blocks, vocabulary, and insights that *tomorrow's* problems will need.

Still, it's foolish to discuss problem solving without actually solving any problems. We need good problems to stimulate our creative juices, to light up our grey cells in a more or less organized fashion, and to make the basic concepts familiar. Mind-bending puzzles are often ideal for this purpose, because they can be presented in a few words, needing no complicated background knowledge.

Václav Havel once remarked that the complexities of life are vast: "There is too much to know... We have to abandon the arrogant belief that the world is merely a puzzle to be solved, a machine with instructions for use waiting to be discovered, a body of information to be fed into a computer." He called

Cretté de Palluel
sheep
Græco-Latin square
–orthogonal latin squares
–latin squares
Langford pairing
Skolem
Steiner triple systems
Groth
multiplication
exact cover problems
apportionment
electoral districts
Havel (playwright and statesman)

for an increased sense of justice and responsibility; for taste, courage, and compassion. His words were filled with great wisdom. Yet thank goodness we do also have puzzles that *can* be solved! Puzzles deserve to be counted among the great pleasures of life, to be enjoyed in moderation like all other treats.

Of course, Langford and Ozanam directed their puzzles to human beings, not to computers. Aren't we missing the point if we merely shuffle such questions off to machines, to be solved by brute force instead of by rational thought? George Brewster, writing to Martin Gardner in 1963, expressed a widely held view as follows: "Feeding a recreational puzzle into a computer is no more than a step above dynamiting a trout stream. Succumbing to instant recreation."

Yes, but that view misses another important point: Simple puzzles often have generalizations that go beyond human ability and arouse our curiosity. The study of those generalizations often suggests instructive methods that apply to numerous other problems and have surprising consequences. Indeed, many of the key techniques that we shall study were born when people were trying to solve various puzzles. While writing this chapter, the author couldn't help relishing the fact that puzzles are now more fun than ever, as computers get faster and faster, because we keep getting more powerful dynamite to play with. [Further comments appear in the author's essay, "Can toy problems be useful?", originally written in 1976; see *Selected Papers on Computer Science* (1996), 169–183.]

Puzzles do have the danger that they can be *too* elegant. Good puzzles tend to be mathematically clean and well-structured, but we also need to learn how to deal systematically with the messy, chaotic, organic stuff that surrounds us every day. Indeed, some computational techniques are important chiefly because they provide powerful ways to cope with such complexities. That is why, for example, the arcane rules of library-card alphabetization were presented at the beginning of Chapter 5, and an actual elevator system was discussed at length to illustrate simulation techniques in Section 2.2.5.

A collection of programs and data called the Stanford GraphBase (SGB) has been prepared so that experiments with combinatorial algorithms can readily be performed on a variety of real-world examples. SGB includes, for example, data about American highways, and an input-output model of the U.S. economy; it records the casts of characters in Homer's *Iliad*, Tolstoy's *Anna Karenina*, and several other novels; it encapsulates the structure of Roget's *Thesaurus* of 1879; it documents hundreds of college football scores; it specifies the gray-value pixels of Leonardo da Vinci's *Gioconda* (Mona Lisa). And perhaps most importantly, SGB contains a collection of five-letter words, which we shall discuss next.

The five-letter words of English. Many of the examples in this chapter will be based on the following list of five-letter words:

$$\texttt{aargh}, \texttt{abaca}, \texttt{abaci}, \texttt{aback}, \texttt{abaft}, \texttt{abase}, \texttt{abash}, \ldots, \texttt{zooms}, \texttt{zowie}. \qquad (8)$$

(There are 5757 words altogether — too many to display here; but those that are missing can readily be imagined.) It's a personal list, collected by the author between 1972 and 1992, beginning when he realized that such words would make `ideal` data for testing many kinds of combinatorial algorithms.

Langford
Ozanam
Brewster
Gardner
Knuth
Stanford GraphBase
SGB
Homer
Tolstoy
novels
Roget
Leonardo da Vinci
five-letter words+++
English+
Knuth+

The list has intentionally been restricted to words that are `truly` part of the English language, in the sense that the author has encountered them in actual use. Unabridged dictionaries contain thousands of entries that are much more esoteric, like `aalii`, `abamp`, ..., `zymin`, and `zyxst`; words like that are useful primarily to Scrabble® players. But unfamiliar words tend to `spoil` the fun for anybody who doesn't know them. Therefore, for twenty years, the author systematically took note of all `words` that seemed `right` for the expository `goals` of *The Art of Computer Programming*.

Finally it was necessary to freeze the collection, in order to have a `fixed point` for reproducible experiments. The English language will always be evolving, but the 5757 SGB words will therefore always stay the same — even though the author has been tempted at times to add a few words that he didn't know in 1992, such as `chads`, `stent`, `blogs`, `ditzy`, `phish`, `bling`, and possibly `tetch`. No; `noway`. The time for any changes to SGB has long since `ended`: `finis`.

The following Glossary is intended to contain all well-known English words ... which may be used in good society, and which can serve as Links. ... There must be a stent to the admission of spick words.
— LEWIS CARROLL, *Doublets: A Word-Puzzle* (1879)

If there is such a verb as to tetch, Mr. Lillywaite tetched.
— ROBERT BARNARD, *Corpse in a Gilded Cage* (1984)

Proper names like `Knuth` are not considered to be legitimate words. But `gauss` and `hardy` are `valid`, because "gauss" is a unit of magnetic induction and "hardy" is hardy. In fact, SGB words are composed entirely of ordinary lowercase letters; the list contains no hyphenated words, contractions, or terms like `blasé` that require an accent. Thus each word can also be regarded as a vector, which has five components in the range $[0\,..\,26)$. In the vector sense, the words `yucca` and `abuzz` are furthest apart: The Euclidean distance between them is

$$\|(24,20,2,2,0)-(0,1,20,25,25)\|_2=\sqrt{24^2+19^2+18^2+23^2+25^2}=\sqrt{2415}.$$

The entire Stanford GraphBase, including all of its programs and data sets, is easy to download from the author's website (see page ii). And the list of all SGB words is even easier to obtain, because it is in the file '`sgb-words.txt`' at the same place. That file contains 5757 lines with one word per line, beginning with '`which`' and ending with '`pupal`'. The words appear in a default order, corresponding to frequency of usage; for example, the words of rank 1000, 2000, 3000, 4000, and 5000 are respectively `ditch`, `galls`, `visas`, `faker`, and `pismo`. The notation '$\mathtt{WORDS}(n)$' will be used in this chapter to stand for the n most common words, according to this ranking.

Incidentally, five-letter words include many plurals of *four-letter words*, and it should be noted that no Victorian-style censorship was done. Potentially offensive vocabulary has been expurgated from *The Official Scrabble® Players Dictionary*, but not from the SGB. One way to ensure that semantically unsuitable

dictionaries
Scrabble®
CARROLL
Lillywaite
BARNARD
Euclidean distance
Stanford GraphBase
Internet
frequency of usage
WORDS(n)++
censorship

terms will not appear in a professional paper based on the SGB wordlist is to restrict consideration to `WORDS`(n) where n is, say, 3000.

Exercises 26–37 below can be used as warmups for initial explorations of the SGB words, which we'll see in many different combinatorial contexts throughout this chapter. For example, while covering problems are still on our minds, we might as well note that the four words '`third flock began jumps`' cover 20 of the first 21 letters of the alphabet. Five words can, however, cover at most 24 different letters, as in {`becks`, `fjord`, `glitz`, `nymph`, `squaw`} — unless we resort to a rare non-SGB word like `waqfs` (Islamic endowments), which can be combined with {`gyved`, `bronx`, `chimp`, `klutz`} to cover 25.

Simple words from `WORDS`(400) suffice to make a *word square*:

```
class
light
agree .                                   (9)
sheep
steps
```

We need to go almost to `WORDS`(3000), however, to obtain a *word cube*,

```
types  yeast  pasta  ester  start
yeast  earth  armor  stove  three
pasta  armor  smoke  token  arena ,       (10)
ester  stove  token  event  rents
start  three  arena  rents  tease
```

in which every 5×5 "slice" is a word square. With a simple extension of the basic dancing links algorithm (see Section 7.2.2.2), one can show after performing about 390 billion mems of computation that `WORDS`(3000) supports only three symmetric word cubes such as (10); exercise 36 reveals the other two. Surprisingly, 83,576 symmetrical cubes can be made from the full set, `WORDS`(5757).

Graphs from words. It's interesting and important to arrange objects into rows, squares, cubes, and other designs; but in practical applications another kind of combinatorial structure is even *more* interesting and important, namely a *graph*. Recall from Section 2.3.4.1 that a graph is a set of points called *vertices*, together with a set of lines called *edges*, which connect certain pairs of vertices. Graphs are ubiquitous, and many beautiful graph algorithms have been discovered, so graphs will naturally be the primary focus of many sections in this chapter. In fact, the Stanford GraphBase is primarily about graphs, as its name implies; and the SGB words were collected chiefly because they can be used to define interesting and instructive graphs.

Lewis Carroll blazed the trail by inventing a game that he called Word-Links or Doublets, at the end of 1877. [See Martin Gardner, *The Universe in a Handkerchief* (1996), Chapter 6.] Carroll's idea, which soon became quite popular, was to transform one word to another by changing a letter at a time:

`tears` — `sears` — `stars` — `stare` — `stale` — `stile` — `smile`. (11)

covering
word square
word cube
dancing links
graphs–
Carroll
Doublets
Gardner
smile

The shortest such transformation is the shortest *path* in a graph, where the vertices of the graph are English words and the edges join pairs of words that have "Hamming distance 1" (meaning that they disagree in just one place).

When restricted to SGB words, Carroll's rule produces a graph of the Stanford GraphBase whose official name is $\mathit{words}(5757,0,0,0)$. Every graph defined by SGB has a unique identifier called its *id*, and the graphs that are derived in Carrollian fashion from SGB words are identified by *id*s of the form $\mathit{words}(n,l,t,s)$. Here n is the number of vertices; l is either 0 or a list of weights, used to emphasize various kinds of vocabulary; t is a threshold so that low-weight words can be disallowed; and s is the seed for any pseudorandom numbers that might be needed to break ties between words of equal weight. The full details needn't concern us, but a few examples will give the general idea:

- $\mathit{words}(n,0,0,0)$ is precisely the graph that arises when Carroll's idea is applied to $\mathtt{WORDS}(n)$, for $1 \le n \le 5757$.
- $\mathit{words}(1000,\{0,0,0,0,0,0,0,0,0\},0,s)$ contains 1000 randomly chosen SGB words, usually different for different values of s.
- $\mathit{words}(766,\{0,0,0,0,0,0,0,1,0\},1,0)$ contains all of the five-letter words that appear in the author's books about TEX and METAFONT.

There are only 766 words in the latter graph, so we can't form very many long paths like (11), although

$$\texttt{basic} — \texttt{basis} — \texttt{bases} — \texttt{based} — \texttt{baked} — \texttt{naked} — \texttt{named} — \texttt{names} — \texttt{games} \qquad (12)$$

is one noteworthy example.

Of course there are many other ways to define the edges of a graph when the vertices represent five-letter words. We could, for example, require the Euclidean distance to be small, instead of the Hamming distance. Or we could declare two words to be adjacent whenever they share a subword of length four; that strategy would substantially enrich the graph, making it possible for `chaos` to yield `peace`, even when confined to the 766 words that are related to TEX:

$$\texttt{chaos} — \texttt{chose} — \texttt{whose} — \texttt{whole} — \texttt{holes} — \texttt{hopes} — \texttt{copes} — \texttt{scope} — \texttt{score} — \texttt{store} — \texttt{stare} — \texttt{spare} — \texttt{space} — \texttt{paces} — \texttt{peace}. \qquad (13)$$

(In this rule we remove a letter, then insert another, possibly in a different place.) Or we might choose a totally different strategy, like putting an edge between word vectors $a_1a_2a_3a_4a_5$ and $b_1b_2b_3b_4b_5$ if and only if their dot product $a_1b_1+a_2b_2+a_3b_3+a_4b_4+a_5b_5$ is a multiple of some parameter m. Graph algorithms thrive on different kinds of data.

SGB words lead also to an interesting family of *directed* graphs, if we write $a_1a_2a_3a_4a_5 \to b_1b_2b_3b_4b_5$ when $\{a_2,a_3,a_4,a_5\} \subseteq \{b_1,b_2,b_3,b_4,b_5\}$ as multisets. (Remove the first letter, insert another, and rearrange.) With this rule we can, for example, transform `words` to `graph` via a shortest oriented path of length six:

$$\texttt{words} \to \texttt{dross} \to \texttt{soars} \to \texttt{orcas} \to \texttt{crash} \to \texttt{sharp} \to \texttt{graph}. \qquad (14)$$

shortest path
Hamming distance
Stanford GraphBase
id
words+
seed
pseudorandom
Carroll
Knuth
TEX
METAFONT
Euclidean distance
Hamming distance
subword
TEX
dot product
directed graphs

Theory is the first term in the Taylor series of practice.
— THOMAS M. COVER (1992)

The number of systems of terminology presently used in graph theory is equal, to a close approximation, to the number of graph theorists.
— RICHARD P. STANLEY (1986)

Graph theory: The basics. A graph G consists of a set V of vertices together with a set E of edges, which are pairs of distinct vertices. We will assume that V and E are *finite* sets unless otherwise specified. We write $u \longrightarrow v$ if u and v are vertices with $\{u,v\} \in E$, and $u \not\longrightarrow v$ if u and v are vertices with $\{u,v\} \notin E$. Vertices with $u \longrightarrow v$ are called "neighbors," and they're also said to be "adjacent" in G. One consequence of this definition is that we have $u \longrightarrow v$ if and only if $v \longrightarrow u$. Another consequence is that $v \not\longrightarrow v$, for all $v \in V$; that is, no vertex is adjacent to itself. (We shall, however, discuss multigraphs below, in which loops from a vertex to itself are permitted.)

The graph $G' = (V', E')$ is a *subgraph* of $G = (V, E)$ if $V' \subseteq V$ and $E' \subseteq E$. It's a *spanning* subgraph of G if, in fact, $V' = V$. And it's an *induced* subgraph of G if E' has as many edges as possible, when V' is a given subset of the vertices. In other words, when $V' \subseteq V$ the subgraph of $G = (V, E)$ induced by V' is $G' = (V', E')$, where

$$E' \;=\; \bigl\{\, \{u,v\} \bigm| u \in V',\ v \in V',\ \text{and } \{u,v\} \in E \,\bigr\}. \tag{15}$$

This subgraph G' is denoted by $G \mid V'$, and often called "G restricted to V'." In the common case where $V' = V \setminus \{v\}$, we write simply $G \setminus v$ ("G minus vertex v") as an abbreviation for $G \mid (V \setminus \{v\})$. The similar notation $G \setminus e$ is used when $e \in E$ to denote the subgraph $G' = (V, E \setminus \{e\})$, obtained by removing an edge instead of a vertex. Notice that all of the SGB graphs known as *words*(n, l, t, s), described earlier, are induced subgraphs of the main graph *words*$(5757, 0, 0, 0)$; only the vocabulary changes in those graphs, not the rule for adjacency.

A graph with n vertices and e edges is said to have *order* n and *size* e. The simplest and most important graphs of order n are the *complete graph* K_n, the *path* P_n, and the *cycle* C_n. Suppose the vertices are $V = \{1, 2, \ldots, n\}$. Then

- K_n has $\binom{n}{2} = \frac{1}{2}n(n-1)$ edges $u \longrightarrow v$ for $1 \le u < v \le n$; every n-vertex graph is a spanning subgraph of K_n.
- P_n has $n-1$ edges $v \longrightarrow (v{+}1)$ for $1 \le v < n$, when $n \ge 1$; it is a path of length $n{-}1$ from 1 to n.
- C_n has n edges $v \longrightarrow ((v \bmod n){+}1)$ for $1 \le v \le n$; it is a graph only when $n = 0$ or $n \ge 3$ (but C_1 and C_2 are multigraphs).

We could actually have defined K_n, P_n, and C_n on the vertices $\{0, 1, \ldots, n{-}1\}$, or on *any* n-element set V instead of $\{1, 2, \ldots, n\}$, because two graphs that differ only in the names of their vertices but not in the structure of their edges are combinatorially equivalent.

Formally, we say that graphs $G = (V, E)$ and $G' = (V', E')$ are *isomorphic* if there is a one-to-one correspondence φ from V to V' such that $u \longrightarrow v$ in G if

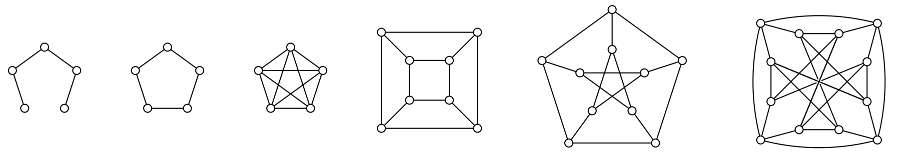
Theory meets practice
Taylor series
COVER
STANLEY
graph theory, introduction to–
neighbors
adjacent
multigraphs
loops
subgraph
spanning
induced
restriction of a graph
notation $G \mid V'$
notation $G \setminus v$
notation $G \setminus e$
order n
size e
complete graph
K_n
path
P_n
cycle
C_n
multigraphs
isomorphic graphs++

and only if $\varphi(u) — \varphi(v)$ in G'. The notation $G \cong G'$ is often used to indicate that G and G' are isomorphic; but we shall often be less precise, by treating isomorphic graphs as if they were equal, and by occasionally writing $G = G'$ even when the vertex sets of G and G' aren't strictly identical.

Small graphs can be defined by simply drawing a diagram, in which the vertices are small circles and the edges are lines between them. Figure 2 illustrates several important examples, whose properties we will be studying later. The Petersen graph in Figure 2(e) is named after Julius Petersen, an early graph theorist who used it to disprove a plausible conjecture [L'*Intermédiaire des Mathématiciens* **5** (1898), 225–227]; it is, in fact, a remarkable configuration that serves as a counterexample to many optimistic predictions about what might be true for graphs in general. The Chvátal graph, Figure 2(f), was introduced by Václav Chvátal in *J. Combinatorial Theory* **9** (1970), 93–94.

notation $G \cong G'$
isomorphic
diagram
Petersen graph
Petersen
Chvátal graph
Chvátal
planar
degree
valency, see degree
regular
cubic
trivalent
Bondy
symmetries
automorphisms

(a) (b) (c) (d) (e) (f)



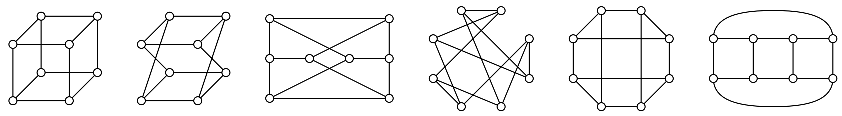
P_5 C_5 K_5 3-cube Petersen graph Chvátal graph

Fig. 2. Six example graphs, which have respectively $(5, 5, 5, 8, 10, 12)$ vertices and $(4, 5, 10, 12, 15, 24)$ edges.

The lines of a graph diagram are allowed to cross each other at points that aren't vertices. For example, the center point of Fig. 2(f) is *not* a vertex of Chvátal's graph. A graph is called *planar* if there's a way to draw it without any crossings. Clearly P_n and C_n are always planar; Fig. 2(d) shows that the 3-cube is also planar. But K_5 has too many edges to be planar (see exercise 46).

The *degree* of a vertex is the number of neighbors that it has. If all vertices have the same degree, the graph is said to be *regular*. In Fig. 2, for example, P_5 is irregular because it has two vertices of degree 1 and three of degree 2. But the other five graphs are regular, of degrees $(2, 4, 3, 3, 4)$ respectively. A regular graph of degree 3 is often called "cubic" or "trivalent."

There are many ways to draw a given graph, some of which are much more perspicuous than others. For example, each of the six diagrams

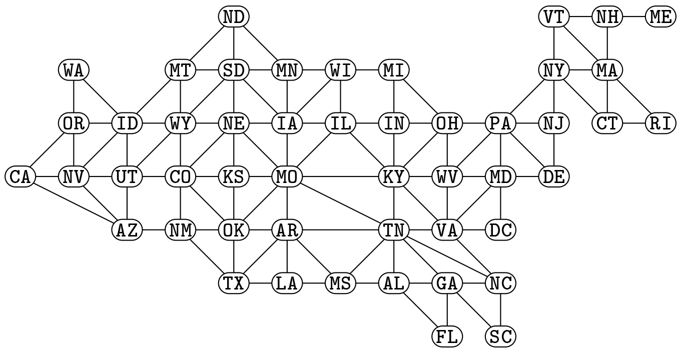


(16)

is isomorphic to the 3-cube, Fig. 2(d). The layout of Chvátal's graph that appears in Fig. 2(f) was discovered by Adrian Bondy many years after Chvátal's paper was published, thereby revealing unexpected symmetries.

The symmetries of a graph, also known as its *automorphisms*, are the permutations of its vertices that preserve adjacency. In other words, the permutation φ is an automorphism of G if we have $\varphi(u) — \varphi(v)$ whenever $u — v$ in G. A well-chosen drawing like Fig. 2(f) can reveal underlying symmetry, but a single diagram isn't always able to display all the symmetries that exist. For example, the 3-cube has 48 automorphisms, and the Petersen graph has 120. We'll study algorithms that deal with isomorphisms and automorphisms in Section 7.2.3. Symmetries can often be exploited to avoid unnecessary computations, making an algorithm almost k times faster when it operates on a graph that has k automorphisms.

Graphs that have evolved in the real world tend to be rather different from the mathematically pristine graphs of Figure 2. For example, here's a familiar graph that has no symmetry whatsoever, although it does have the virtue of being planar:



(17)

It represents the contiguous United States of America, and we'll be using it later in several examples. The 49 vertices of this diagram have been labeled with two-letter postal codes for convenience, instead of being reduced to empty circles.

Paths and cycles. A spanning path of a graph is called a *Hamiltonian path*, and a spanning cycle is called a *Hamiltonian cycle*, because W. R. Hamilton invented and sold a puzzle in 1859 whose goal was to find such paths and cycles on the edges of a dodecahedron. T. P. Kirkman had independently studied the problem for polyhedra in general, in *Philosophical Transactions* **148** (1858), 145–161. [See *Graph Theory 1736–1936* by N. L. Biggs, E. K. Lloyd, and R. J. Wilson (1998), Chapter 2.] The task of finding a spanning path or cycle is, however, much older — indeed, we can legitimately consider it to be the oldest combinatorial problem of all, because paths and tours of a knight on a chessboard have a continuous history going back to ninth-century India (see Section 7.3.3). A graph is called *Hamiltonian* if it has a Hamiltonian cycle. (The Petersen graph, incidentally, is the smallest 3-regular graph that is neither planar nor Hamiltonian; see C. de Polignac, *Bull. Soc. Math. de France* **27** (1899), 142–145.)

The *girth* of a graph is the length of its shortest cycle; the girth is infinite if the graph is acyclic (containing no cycles). For example, the six graphs of Fig. 2 have girths $(\infty, 5, 3, 4, 5, 4)$, respectively. It's not difficult to prove that a graph of minimum degree k and girth 5 must have at least $k^2 + 1$ vertices. Further analysis shows in fact that this minimum value is achievable only if $k = 2$ (C_5), $k = 3$ (Petersen), $k = 7$, or perhaps $k = 57$. (See exercises 63 and 65.)

planar
contiguous United States of America
postal codes
spanning
Hamiltonian path
Hamiltonian cycle
Hamilton
dodecahedron
Kirkman
Biggs
Lloyd
Wilson
knight
chessboard
Hamiltonian
Petersen graph
de Polignac
girth
acyclic

The *distance* $d(u,v)$ between two vertices u and v is the minimum length of a path from u to v in the graph; it is infinite if there's no such path. Clearly $d(v,v)=0$, and $d(u,v)=d(v,u)$. We also have the triangle inequality

$$d(u,v)+d(v,w)\ \ge\ d(u,w). \tag{18}$$

For if $d(u,v)=p$ and $d(v,w)=q$ and $p<\infty$ and $q<\infty$, there are paths

$$u=u_0 - u_1 - \cdots - u_p=v \quad\text{and}\quad v=v_0 - v_1 - \cdots - v_q=w, \tag{19}$$

and we can find the least subscript r such that $u_r=v_s$ for some s. Then

$$u_0 - u_1 - \cdots - u_{r-1} - v_s - v_{s+1} - \cdots - v_q \tag{20}$$

is a path of length $\le p+q$ from u to w.

The *diameter* of a graph is the maximum of $d(u,v)$, over all vertices u and v. The graph is *connected* if its diameter is finite. The vertices of a graph can always be partitioned into connected *components*, where two vertices u and v belong to the same component if and only if $d(u,v)<\infty$.

In the graph *words*$(5757,0,0,0)$, for example, we have $d(\texttt{tears},\texttt{smile})=6$, because (11) is a shortest path from `tears` to `smile`. Also $d(\texttt{tears},\texttt{happy})=6$, and $d(\texttt{smile},\texttt{happy})=10$, and $d(\texttt{world},\texttt{court})=6$. But $d(\texttt{world},\texttt{happy})=\infty$; the graph isn't connected. In fact, it contains 671 words like `aloof`, which have no neighbors and form connected components of order 1 all by themselves. Word pairs such as `alpha` — `aloha`, `droid` — `druid`, and `opium` — `odium` account for 103 further components of order 2. Some components of order 3, like `chain` — `chair` — `choir`, are paths; others, like $\{\texttt{getup},\texttt{letup},\texttt{setup}\}$, are cycles. A few more small components are also present, like the curious path

$$\texttt{login} - \texttt{logic} - \texttt{yogic} - \texttt{yogis} - \texttt{yogas} - \texttt{togas}, \tag{21}$$

whose words have no other neighbors. But the vast majority of all five-letter words belong to a giant component of order 4493. If you can go two steps away from a given word, the odds are better than 15 to 1 that your word is connected to everything in the giant component.

Similarly, the graph *words*$(n,0,0,0)$ has a giant component of order (3825, 2986, 2056, 1198, 224) when $n=(5000,4000,3000,2000,1000)$, respectively. But if n is small, there aren't enough edges to provide much connectivity. For example, *words*$(500,0,0,0)$ has 327 different components, none of order 15 or more.

The concept of distance can be generalized to $d(v_1,v_2,\ldots,v_k)$ for any value of k, meaning the minimum number of edges in a connected subgraph that contains the vertices $\{v_1,v_2,\ldots,v_k\}$. For example, $d(\texttt{blood},\texttt{sweat},\texttt{tears})$ turns out be 15, because the subgraph

$$\begin{array}{ccccccccccccc}
\texttt{blood} & - & \texttt{brood} & - & \texttt{broad} & - & \texttt{bread} & - & \texttt{tread} & - & \texttt{treed} & - & \texttt{tweed}\\
 & & & & & & & & & & | & & |\\
\texttt{tears} & - & \texttt{teams} & - & \texttt{trams} & - & \texttt{trims} & - & \texttt{tries} & - & \texttt{trees} & & \texttt{tweet}\\
 & & & & & & & & & & & & |\\
 & & & & & & & & & & \texttt{sweat} & - & \texttt{sweet}
\end{array} \tag{22}$$

has 15 edges, and there's no suitable 14-edge subgraph.

distance
shortest path
triangle inequality
diameter
connected
components
5-letter wds
smile
giant component
distance, generalized+

We noted in Section 2.3.4.1 that a connected graph with fewest edges is called a *free tree*. A subgraph that corresponds to the generalized distance $d(v_1, \ldots, v_k)$ will always be a free tree. It is misleadingly called a *Steiner tree,* because Jacob Steiner once mentioned the case $k = 3$ for points $\{v_1, v_2, v_3\}$ in the Euclidean plane [*Crelle* **13** (1835), 362–363]. Franz Heinen had solved that problem in *Über Systeme von Kräften* (1834); Gauss extended the analysis to $k = 4$ in a letter to Schumacher (21 March 1836).

Coloring. A graph is said to be *k-partite* or *k-colorable* if its vertices can be partitioned into k or fewer parts, with the endpoints of each edge belonging to different parts — or equivalently, if there's a way to paint its vertices with at most k different colors, never assigning the same color to two adjacent vertices. The famous Four Color Theorem, conjectured by F. Guthrie in 1852 and finally proved with massive computer aid by K. Appel, W. Haken, and J. Koch [*Illinois J. Math.* **21** (1977), 429–567], states that *every planar graph is 4-colorable.* No simple proof is known, but special cases like (17) can be colored at sight (see exercise 45); and $O(n^2)$ steps suffice to 4-color a planar graph in general [N. Robertson, D. P. Sanders, P. Seymour, and R. Thomas, *STOC* **28** (1996), 571–575].

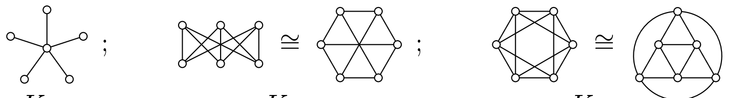
The case of 2-colorable graphs is especially important in practice. A 2-partite graph is generally called *bipartite*, or simply a "bigraph"; every edge of such a graph has one endpoint in each part.

Theorem B. *A graph is bipartite if and only if it contains no cycle of odd length.*

Proof. [See D. Kőnig, *Math. Annalen* **77** (1916), 453–454.] Every subgraph of a k-partite graph is k-partite. Therefore the cycle C_n can be a subgraph of a bipartite graph only if C_n itself is a bigraph, in which case n must be even.

Conversely, if a graph contains no odd cycles we can color its vertices with the two colors $\{0, 1\}$ by carrying out the following procedure: Begin with all vertices uncolored. If all neighbors of colored vertices are already colored, choose an uncolored vertex w, and color it 0. Otherwise choose a colored vertex u that has an uncolored neighbor v; assign to v the opposite color. Exercise 48 proves that a valid 2-coloring is eventually obtained. ▌

The *complete bipartite graph* $K_{m,n}$ is the largest bipartite graph whose vertices have two parts of sizes m and n. We can define it on the vertex set $\{1, 2, \ldots, m+n\}$ by saying that $u \text{ — } v$ whenever $1 \le u \le m < v \le m + n$. In other words, $K_{m,n}$ has mn edges, one for each way to choose one vertex in the first part and another in the second part. Similarly, the *complete k-partite graph* $K_{n_1,\ldots,n_k}$ has $N = n_1 + \cdots + n_k$ vertices partitioned into parts of sizes $\{n_1, \ldots, n_k\}$, and it has edges between any two vertices that don't belong to the same part. Here are some examples when $N = 6$:



$K_{1,5}$; $K_{3,3}$; $K_{2,2,2}$. (23)

Notice that $K_{1,n}$ is a free tree; it is popularly called the *star graph* of order $n+1$.

free tree
Steiner tree
Steiner
Euclidean plane
Heinen
Gauss
Schumacher
Coloring
k-partite
k-colorable
Four Color Theorem
Guthrie
Appel
Haken
Koch
planar
Robertson
Sanders
Seymour
Thomas
bipartite
bigraph
Kőnig
subgraph
complete bipartite graph
complete k-partite graph
free tree
star graph

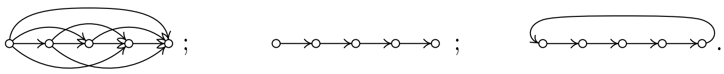
From now on say "digraph" instead of "directed graph."
It is clear and short and it will catch on.
— GEORGE PÓLYA, letter to Frank Harary (c. 1954)

Directed graphs. In Section 2.3.4.2 we defined *directed graphs* (or *digraphs*), which are very much like graphs except that they have *arcs* instead of edges. An arc $u \longrightarrow v$ runs from one vertex to another, while an edge $u \;\frac{\quad}{}\; v$ joins two vertices without distinguishing between them. Furthermore, digraphs are allowed to have self-loops $v \longrightarrow v$ from a vertex to itself, and more than one arc $u \longrightarrow v$ may be present between the same vertices u and v.

Formally, a digraph $D = (V, A)$ of order n and size m is a set V of n vertices and a multiset A of m ordered pairs (u, v), where $u \in V$ and $v \in V$. The ordered pairs are called arcs, and we write $u \longrightarrow v$ when $(u, v) \in A$. The digraph is called *simple* if A is actually a set instead of a general multiset — namely, if there's at most one arc (u, v) for all u and v. Each arc (u, v) has an initial vertex u and a final vertex v, also called its "tip." Each vertex has an *out-degree* $d^+(v)$, the number of arcs for which v is the initial vertex, and an *in-degree* $d^-(v)$, the number of arcs for which v is the tip. A vertex with in-degree 0 is called a "source"; a vertex with out-degree 0 is called a "sink." Notice that $\sum_{v \in V} d^+(v) = \sum_{v \in V} d^-(v)$, because both sums are equal to m, the total number of arcs.

Most of the notions we've defined for graphs carry over to digraphs in a natural way, if we just insert the word "directed" or "oriented" (or the syllable "di") when it's necessary to distinguish between edges and arcs. For example, digraphs have subdigraphs, which can be spanning or induced or neither. An isomorphism between digraphs $D = (V, A)$ and $D' = (V', A')$ is a one-to-one correspondence φ from V to V' for which the number of arcs $u \longrightarrow v$ in D equals the number of arcs $\varphi(u) \longrightarrow \varphi(v)$ in D', for all $u, v \in V$.

Diagrams for digraphs use arrows between the vertices, instead of unadorned lines. The simplest and most important digraphs of order n are directed variants of the graphs K_n, P_n, and C_n, namely the *transitive tournament* $K_n^{\rightarrow}$, the *oriented path* $P_n^{\rightarrow}$, and the *oriented cycle* $C_n^{\rightarrow}$. They can be schematically indicated by the following diagrams for $n = 5$:

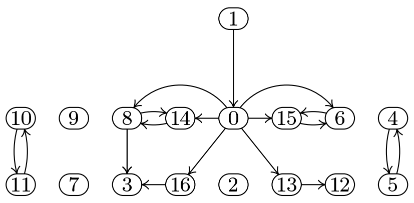


$K_5^{\rightarrow}$; $P_5^{\rightarrow}$; $C_5^{\rightarrow}$. (24)

There's also the *complete digraph* J_n, which is the largest simple digraph on n vertices; it has n^2 arcs $u \longrightarrow v$, one for each choice of u and v.

Figure 3 shows a more elaborate diagram, for a digraph of order 17 that we might call "expressly oriented": It is the directed graph described by Hercule Poirot in Agatha Christie's novel *Murder on the Orient Express* (1934). Vertices correspond to the berths of the Stamboul–Calais coach in that story, and an arc $u \longrightarrow v$ means that the occupant of berth u has corroborated the alibi of the person in berth v. This example has six connected components, namely $\{0, 1, 3, 6, 8, 12, 13, 14, 15, 16\}$, $\{2\}$, $\{4, 5\}$, $\{7\}$, $\{9\}$, and $\{10, 11\}$, because connectivity in a digraph is determined by treating arcs as edges.

PÓLYA
Harary
directed graphs–
arcs–
arcs
self-loops
order
size
multiset
simple
initial vertex
final vertex
tip
out-degree
$d^+(v)$
in-degree
$d^-(v)$
source
sink
spanning
induced
isomorphism
Diagrams for digraphs
transitive tournament
$K_n^{\rightarrow}$
oriented path
$P_n^{\rightarrow}$
oriented cycle
$C_n^{\rightarrow}$
complete digraph
J_n
Poirot
Christie
components
connectivity in a digraph
arcs as edges



LEGEND
0: Pierre Michel, the French conductor
1: Hercule Poirot, the Belgian detective
2: Samuel Edward Ratchett, the deceased American
3: Caroline Martha Hubbard, the American matron
4: Edward Henry Masterman, the British valet
5: Antonio Foscarelli, the Italian automobile salesman
6: Hector MacQueen, the American secretary
7: Harvey Harris, the Englishman who didn't show up
8: Hildegarde Schmidt, the German lady's maid
9: (vacancy)
10: Greta Ohlsson, the Swedish nurse
11: Mary Hermione Debenham, the English governess
12: Helena Maria Andrenyi, the beautiful countess
13: Rudolph Andrenyi, the Hungarian count/diplomat
14: Natalia Dragomiroff, the Russian princess dowager
15: Colonel Arbuthnot, the British officer from India
16: Cyrus Bettman Hardman, the American detective

Fig. 3. A digraph of order 17 and size 18, devised by Agatha Christie.

Two arcs are *consecutive* if the tip of the first is the initial vertex of the second. A sequence of consecutive arcs $(a_1, a_2, \ldots, a_k)$ is called a *walk* of length k; it can be symbolized by showing the vertices as well as the arcs:

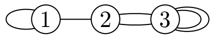
$$v_0 \xrightarrow{a_1} v_1 \xrightarrow{a_2} v_2 \cdots v_{k-1} \xrightarrow{a_k} v_k. \qquad (25)$$

In a simple digraph it's sufficient merely to specify the vertices; for example, $1 \longrightarrow 0 \longrightarrow 8 \longrightarrow 14 \longrightarrow 8 \longrightarrow 3$ is a walk in Fig. 3. The walk in (25) is an oriented path when the vertices $\{v_0, v_1, \ldots, v_k\}$ are distinct; it's an oriented cycle when they are distinct except that $v_k = v_0$.

In a digraph, the directed distance $d(u,v)$ is the number of arcs in the shortest *oriented* path from u to v, which is also the length of the shortest walk from u to v. It may differ from $d(v,u)$; but the triangle inequality (18) remains valid.

Every graph can be regarded as a digraph, because an edge u — v is essentially equivalent to a matched pair of arcs, $u \longrightarrow v$ and $v \longrightarrow u$. The digraph obtained in this way retains all the properties of the original graph; for example, the degree of each vertex in the graph becomes its out-degree in the digraph, and also its in-degree in the digraph. Furthermore, distances remain the same.

A *multigraph* (V,E) is like a graph except that its edges E can be any *multiset* of pairs $\{u,v\}$; edges v — v that loop from a vertex to itself, which correspond to "multipairs" $\{v,v\}$, are also permitted. For example,



(26)

is a multigraph of order 3 with six edges, $\{1,1\}$, $\{1,2\}$, $\{2,3\}$, $\{2,3\}$, $\{3,3\}$, and $\{3,3\}$. The vertex degrees in this example are $d(1) = d(2) = 3$ and $d(3) = 6$, because each loop contributes 2 to the degree of its vertex. An edge loop v — v becomes *two* arc loops $v \longrightarrow v$ when a multigraph is regarded as a digraph.

Representation of graphs and digraphs. Any digraph, and therefore any graph or multigraph, is completely described by its *adjacency matrix* $A = (a_{uv})$, which has n rows and n columns when there are n vertices. Each entry a_{uv} of this matrix specifies the number of arcs from u to v. For example, the adjacency matrices for $K_3^{\rightarrow}$, $P_3^{\rightarrow}$, $C_3^{\rightarrow}$, J_3, and (26) are respectively

$$K_3^{\rightarrow} = \begin{pmatrix} 011 \\ 001 \\ 000 \end{pmatrix}, \quad P_3^{\rightarrow} = \begin{pmatrix} 010 \\ 001 \\ 000 \end{pmatrix}, \quad C_3^{\rightarrow} = \begin{pmatrix} 010 \\ 001 \\ 100 \end{pmatrix}, \quad J_3 = \begin{pmatrix} 111 \\ 111 \\ 111 \end{pmatrix}, \quad A = \begin{pmatrix} 210 \\ 102 \\ 024 \end{pmatrix}. \qquad (27)$$

consecutive
walk
oriented path
oriented cycle
directed distance $d(u,v)$
triangle inequality
graph can be regarded as a digraph
edges as arcs
out-degree
in-degree
multigraph
loop
multipairs
parallel (repeated) edges
degrees
–graph theory, introduction to
adjacency matrix+

matrix multiplication
walks
permutation matrix
0–1 matrix
sparse
Stanford GraphBase

The powerful mathematical tools of matrix theory make it possible to prove many nontrivial results about graphs by studying their adjacency matrices; exercise 65 provides a particularly striking example of what can be done. One of the main reasons is that matrix multiplication has a simple interpretation in the context of digraphs. Consider the square of A, where the element in row u and column v is

$$(A^2)_{uv} \;=\; \sum_{w\in V} a_{uw}a_{wv}, \tag{28}$$

by definition. Since a_{uw} is the number of arcs from u to w, we see that $a_{uw}a_{wv}$ is the number of walks of the form $u \longrightarrow w \longrightarrow v$. Therefore $(A^2)_{uv}$ is the total number of walks of length 2 from u to v. Similarly, the entries of A^k tell us the total number of walks of length k between any ordered pair of vertices, for all $k \geq 0$. For example, the matrix A in (27) satisfies

$$A = \begin{pmatrix} 2 & 1 & 0 \\ 1 & 0 & 2 \\ 0 & 2 & 4 \end{pmatrix}, \qquad A^2 = \begin{pmatrix} 5 & 2 & 2 \\ 2 & 5 & 8 \\ 2 & 8 & 20 \end{pmatrix}, \qquad A^3 = \begin{pmatrix} 12 & 9 & 12 \\ 9 & 18 & 42 \\ 12 & 42 & 96 \end{pmatrix}; \tag{29}$$

there are 12 walks of length 3 from the vertex 1 of the multigraph (26) to vertex 3, and 18 such walks from vertex 2 to itself.

Reordering of the vertices changes an adjacency matrix from A to P^-AP, where P is a permutation matrix (a 0–1 matrix with exactly one 1 in each row and column), and $P^- = P^T$ is the matrix for the inverse permutation. Thus

$$\begin{pmatrix} 210 \\ 102 \\ 024 \end{pmatrix}, \quad \begin{pmatrix} 201 \\ 042 \\ 120 \end{pmatrix}, \quad \begin{pmatrix} 012 \\ 120 \\ 204 \end{pmatrix}, \quad \begin{pmatrix} 021 \\ 240 \\ 102 \end{pmatrix}, \quad \begin{pmatrix} 402 \\ 021 \\ 210 \end{pmatrix}, \quad \text{and} \quad \begin{pmatrix} 420 \\ 201 \\ 012 \end{pmatrix} \tag{30}$$

are all adjacency matrices for (26), and there are no others.

There are more than $2^{n(n-1)/2}/n!$ graphs of order n, when $n > 1$, and almost all of them require $\Omega(n^2)$ bits of data in their most economical encoding. Consequently the best way to represent the vast majority of all possible graphs inside a computer, from the standpoint of memory usage, is essentially to work with their adjacency matrices.

But the graphs that actually arise in practical problems have quite different characteristics from graphs that are chosen at random from the set of all possibilities. A real-life graph usually turns out to be "sparse," having say $O(n \log n)$ edges instead of $\Omega(n^2)$, unless n is rather small, because $\Omega(n^2)$ bits of data are difficult to generate. For example, suppose the vertices correspond to people, and the edges correspond to friendships. If we consider 5 billion people, few of them will have more than 10000 friends. But even if everybody had 10000 friends, on average, the graph would still have only 2.5×10^{13} edges, while almost all graphs of order 5 billion have approximately 6.25×10^{18} edges.

Thus the best way to represent a graph inside a machine usually turns out to be rather different than to record n^2 values a_{uv} of adjacency matrix elements. Instead, the algorithms of the Stanford GraphBase were developed with a data structure akin to the linked representation of sparse matrices discussed in Section 2.2.6, though somewhat simplified. That approach has proved to be not only versatile and efficient, but also easy to use.

The SGB representation of a digraph is a combination of sequential and linked allocation, using nodes of two basic types. Some nodes represent vertices, other nodes represent arcs. (There's also a third type of node, which represents an entire graph, for algorithms that deal with several graphs at once. But each graph needs only one graph node, so the vertex and arc nodes predominate.)

Here's how it works: Every SGB digraph of order n and size m is built upon a sequential array of n vertex nodes, making it easy to access vertex k for $0 \leq k < n$. The m arc nodes, by contrast, are linked together within a general memory pool that is essentially unstructured. Each vertex node typically occupies 32 bytes, and each arc node occupies 20 (and the graph node occupies 220); but the node sizes can be modified without difficulty. A few fields of each node have a fixed, definite meaning in all cases; the remaining fields can be used for different purposes in different algorithms or in different phases of a single algorithm. The fixed-purpose parts of a node are called its "standard fields," and the multipurpose parts are called its "utility fields."

Every vertex node has two standard fields called `NAME` and `ARCS`. If v is a variable that points to a vertex node, we'll call it a *vertex variable.* Then `NAME`(v) points to a string of characters that can be used to identify the corresponding vertex in human-oriented output; for example, the 49 vertices of graph (17) have names like `CA`, `WA`, `OR`, ..., `RI`. The other standard field, `ARCS`(v), is far more important in algorithms: It points to an arc node, the first in a singly linked list of length $d^+(v)$, with one node for each arc that emanates from vertex v.

Every arc node has two standard fields called `TIP` and `NEXT`; a variable a that points to an arc node is called an *arc variable.* `TIP`(a) points to the vertex node that represents the tip of arc a; `NEXT`(a) points to the arc node that represents the next arc whose initial vertex agrees with that of a.

A vertex v with out-degree 0 is represented by letting `ARCS`$(v) = \Lambda$ (the null pointer). Otherwise if, say, the out-degree is 3, the data structure contains three arc nodes with `ARCS`$(v) = a_1$, `NEXT`$(a_1) = a_2$, `NEXT`$(a_2) = a_3$, and `NEXT`$(a_3) = \Lambda$; and the three arcs from v lead to `TIP`(a_1), `TIP`(a_2), `TIP`(a_3).

Suppose, for example, that we want to compute the out-degree of vertex v, and store it in a utility field called `ODEG`. It's easy:

$$\begin{aligned}&\text{Set } a \leftarrow \texttt{ARCS}(v) \text{ and } d \leftarrow 0.\\ &\text{While } a \neq \Lambda, \text{ set } d \leftarrow d+1 \text{ and } a \leftarrow \texttt{NEXT}(a).\\ &\text{Set } \texttt{ODEG}(v) \leftarrow d.\end{aligned} \tag{31}$$

When a graph or a multigraph is considered to be a digraph, as mentioned above, its edges $u \mathrel{-\!\!-} v$ are each equivalent to two arcs, $u \longrightarrow v$ and $v \longrightarrow u$. These arcs are called "mates"; and they occupy two arc nodes, say a and a', where a appears in the list of arcs from u and a' appears in the list of arcs from v. Then `TIP`$(a) = v$ and `TIP`$(a') = u$. We'll also write

$$\texttt{MATE}(a) = a' \qquad \text{and} \qquad \texttt{MATE}(a') = a, \tag{32}$$

in algorithms that want to move rapidly from one list to another. However, we usually won't need to store an explicit pointer from an arc to its mate, or to have

SGB representation of a digraph
sequential
linked allocation
node sizes+
standard fields
utility fields
NAME
ARCS
vertex variable
TIP
NEXT
arc variable
out-degree
multigraph
edges as arcs+
MATE+
mates+
MATE(a)+

a utility field called `MATE` within each arc node, because the necessary link can be deduced *implicitly* when the data structure has been constructed cleverly.

The implicit-mate trick works like this: While creating each edge $u \longrightarrow v$ of an undirected graph or multigraph, we introduce *consecutive* arc nodes for $u \longrightarrow v$ and $v \longrightarrow u$. For example, if there are 20 bytes per arc node, we'll reserve 40 consecutive bytes for each new pair. We can also make sure that the memory address of the first byte is a multiple of 8. Then if the arc node a is in memory location α, its mate is in location

$$\left\{\begin{matrix} \alpha + 20, & \text{if } \alpha \bmod 8 = 0 \\ \alpha - 20, & \text{if } \alpha \bmod 8 = 4 \end{matrix}\right\} \;=\; \alpha - 20 + \bigl(40 \,\&\, ((\alpha \,\&\, 4) - 1)\bigr). \qquad (33)$$

Such tricks are valuable in combinatorial problems, when operations might be performed a trillion times, because every way to save 3.6 nanoseconds per operation will make such a computation finish an hour sooner. But (33) isn't directly "portable" from one implementation to another. If the size of an arc node were changed from 20 to 24, for example, we would have to change the numbers 40, 20, 8, and 4 in (33) to 48, 24, 16, and 8.

The algorithms in this book will make no assumptions about node sizes. Instead, we'll adopt a convention of the C programming language and its descendants, so that if a points to an arc node, '$a+1$' denotes a pointer to the arc node that follows it in memory. And in general

$$\mathtt{LOC(NODE(}a+k\mathtt{))} \;=\; \mathtt{LOC(NODE(}a\mathtt{))} + kc, \qquad (34)$$

when there are c bytes in each arc node. Similarly, if v is a vertex variable, '$v+k$' will stand for the kth vertex node following node v; the actual memory location of that node will be v plus k times the size of a vertex node.

The standard fields of a graph node g include `M(`g`)`, the total number of arcs; `N(`g`)`, the total number of vertices; `VERTICES(`g`)`, a pointer to the first vertex node in the sequential list of all vertex nodes; `ID(`g`)`, the graph's identification, which is a string like `words(5757,0,0,0)`; and some other fields needed for the allocation and recycling of memory when the graph grows or shrinks, or for exporting a graph to external formats that interface with other users and other graph-manipulation systems. But we will rarely need to refer to any of these graph node fields, nor will it be necessary to give a complete description of SGB format here, since we shall describe almost all of the graph algorithms in this chapter by sticking to an English-language description at a fairly abstract level instead of descending to the bit level of machine programs.

A simple graph algorithm. To illustrate a medium-high-level algorithm of the kind that will appear later, let's convert the proof of Theorem B into a step-by-step procedure that paints the vertices of a given graph with two colors whenever that graph is bipartite.

Algorithm B (*Bipartiteness testing*). Given a graph represented in SGB format, this algorithm either finds a 2-coloring with `COLOR(`v`)` $\in \{0,1\}$ in each vertex v, or it terminates unsuccessfully when no valid 2-coloring is possible. Here `COLOR` is a utility field in each vertex node. Another vertex utility field, `LINK(`v`)`, is a

implicitly
bitwise AND
portable
C
LOC
graph node
M(g)
N(g)
VERTICES(g)
ID(g)
id
–directed graphs
bipartiteness testing+

vertex pointer used to maintain a stack of all colored vertices whose neighbors have not yet been examined. An auxiliary vertex variable s points to the top of this stack. The algorithm also uses variables u, v, w for vertices and a for arcs. The vertex nodes are assumed to be $v_0 + k$ for $0 \le k < n$.

B1. [Initialize.] Set $\mathtt{COLOR}(v_0+k) \leftarrow -1$ for $0 \le k < n$. (Now all vertices are uncolored.) Then set $w \leftarrow v_0 + n$.

B2. [Done?] (At this point all vertices $\ge w$ have been colored, and so have the neighbors of all colored vertices.) Terminate the algorithm successfully if $w = v_0$. Otherwise set $w \leftarrow w - 1$, the next lower vertex node.

B3. [Color w if necessary.] If $\mathtt{COLOR}(w) \ge 0$, return to B2. Otherwise set $\mathtt{COLOR}(w) \leftarrow 0$, $\mathtt{LINK}(w) \leftarrow \Lambda$, and $s \leftarrow w$.

B4. [Stack $\Rightarrow u$.] Set $u \leftarrow s$, $s \leftarrow \mathtt{LINK}(s)$, $a \leftarrow \mathtt{ARCS}(u)$. (We will examine all neighbors of the colored vertex u.)

B5. [Done with u?] If $a = \Lambda$, go to B8. Otherwise set $v \leftarrow \mathtt{TIP}(a)$.

B6. [Process v.] If $\mathtt{COLOR}(v) < 0$, set $\mathtt{COLOR}(v) \leftarrow 1 - \mathtt{COLOR}(u)$, $\mathtt{LINK}(v) \leftarrow s$, and $s \leftarrow v$. Otherwise if $\mathtt{COLOR}(v) = \mathtt{COLOR}(u)$, terminate unsuccessfully.

B7. [Loop on a.] Set $a \leftarrow \mathtt{NEXT}(a)$ and return to B5.

B8. [Stack nonempty?] If $s \ne \Lambda$, return to B4. Otherwise return to B2. ▮

This algorithm is a variant of a general graph traversal procedure called "depth-first search," which we will study in detail in Section 7.4.1. Its running time is $O(m + n)$ when there are m arcs and n vertices (see exercise 70); therefore it is well adapted to the common case of sparse graphs. With small changes we can make it output an odd-length cycle whenever it terminates unsuccessfully, thereby proving the impossibility of a 2-coloring (see exercise 72).

Examples of graphs. The Stanford GraphBase includes a library of more than three dozen generator routines, capable of producing a great variety of graphs and digraphs for use in experiments. We've already discussed *words*; now let's look at a few of the others, in order to get a feeling for some of the possibilities.

• *roget*$(1022, 0, 0, 0)$ is a directed graph with 1022 vertices and 5075 arcs. The vertices represent the categories of words or concepts that P. M. Roget and J. L. Roget included in their famous 19th-century *Thesaurus* (London: Longmans, Green, 1879). The arcs are the cross references between categories, as found in that book. For example, typical arcs are `water` ⟶ `moisture`, `discovery` ⟶ `truth`, `preparation` ⟶ `learning`, `vulgarity` ⟶ `ugliness`, `wit` ⟶ `amusement`.

• *book*(`"jean"`$, 80, 0, 1, 356, 0, 0, 0)$ is a graph with 80 vertices and 254 edges. The vertices represent the characters of Victor Hugo's *Les Misérables*; the edges connect characters who encounter each other in that novel. Typical edges are `Fantine` — `Javert`, `Cosette` — `Thénardier`.

• *bi_book*(`"jean"`$, 80, 0, 1, 356, 0, 0, 0)$ is a bipartite graph with 80+356 vertices and 727 edges. The vertices represent characters or chapters in *Les Misérables*; the edges connect characters with the chapters in which they appear (for instance, `Napoleon` — `2.1.8`, `Marius` — `4.14.4`).

stack
vertex variable
arc variable
depth-first search
sparse graphs
–arcs
Stanford GraphBase++
generator routines
roget
Roget
Roget
book
Hugo
novel
bi_book
bipartite graph

• *plane_miles*$(128, 0, 0, 0, 1, 0, 0)$ is a planar graph with 129 vertices and 381 edges. The vertices represent 128 cities in the United States or Canada, plus a special vertex `INF` for a "point at infinity." The edges define the so-called *Delaunay triangulation* of those cities, based on latitude and longitude in a plane; this means that u — v if and only if the smallest circle that passes through u and v does not enclose any other vertex. Edges also run between `INF` and all vertices that lie on the convex hull of all city locations. Typical edges are `Seattle, WA` — `Vancouver, BC` — `INF`; `Toronto, ON` — `Rochester, NY`.

• *plane_lisa*$(360, 250, 15, 0, 360, 0, 250, 0, 0, 2295000)$ is a planar graph that has 3027 vertices and 5967 edges. It is obtained by starting with a digitized image of Leonardo da Vinci's *Mona Lisa*, having 360 rows and 250 columns of pixels, then rounding the pixel intensities to 16 levels of gray from 0 (black) to 15 (white). The resulting 3027 rookwise connected regions of constant brightness are then considered to be neighbors when they share a pixel boundary. (See Fig. 4.)

plane_miles
planar graph
infinity
Delaunay triangulation
convex hull
plane_lisa
digitized
image
Leonardo da Vinci
Mona Lisa
pixels
rookwise connected
bi_lisa
bipartite graph
smile
Gherardini, Lisa
raman
expander graph
regular graph+
Ramanujan graphs
girth
diameter

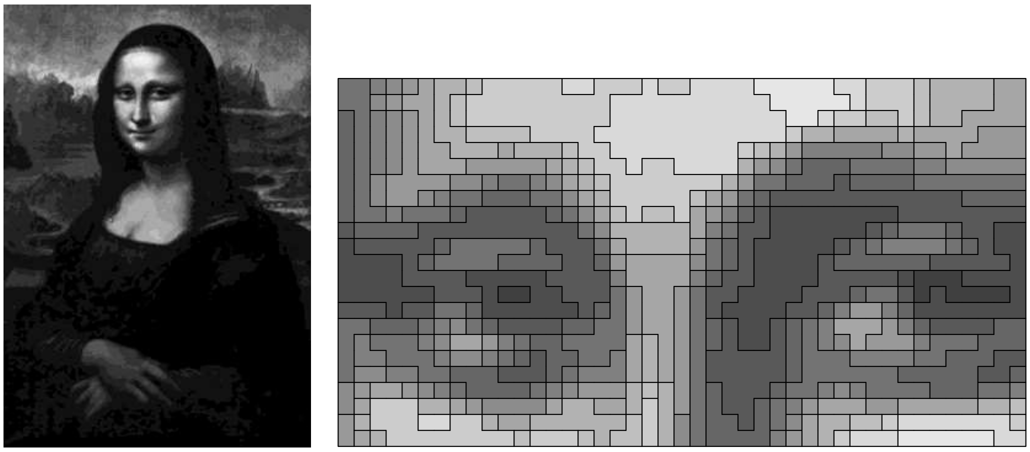


Fig. 4. A digital rendition of *Mona Lisa*, with a closeup detail (best viewed from afar).

• *bi_lisa*$(360, 250, 0, 360, 0, 250, 8192, 0)$ is a bipartite graph with $360 + 250 = 610$ vertices and 40923 edges. It's another takeoff on Leonardo's famous painting, this time linking rows and columns where the brightness level is at least 1/8. For example, the edge `r102` — `c113` occurs right in the middle of Lisa's "smile."

• *raman*$(31, 23, 3, 1)$ is a graph with quite a different nature from the SGB graphs in previous examples. Instead of being linked to language, literature, or other outgrowths of human culture, it's a so-called "Ramanujan expander graph," based on strict mathematical principles. Each of its $(23^3 - 23)/2 = 6072$ vertices has degree 32; hence it has 97152 edges. The vertices correspond to equivalence classes of 2×2 matrices that are nonsingular modulo 23; a typical edge is `(2,7;1,1)` — `(4,6;1,3)`. Ramanujan graphs are important chiefly because they have unusually high girth and low diameter for their size and degree. This one has girth 4 and diameter 4.

• *raman*$(5,37,4,1)$, similarly, is a regular graph of degree 6 with 50616 vertices and 151848 edges. It has girth 10, diameter 10, and happens also to be bipartite.

• *random_graph*$(1000,5000,0,0,0,0,0,0,0,s)$ is a graph with 1000 vertices, 5000 edges, and seed s. It "evolved" by starting with no edges, then by repeatedly choosing pseudorandom vertex numbers $0 \le u,v < 1000$ and adding the edge $u — v$, unless $u = v$ or that edge was already present. When $s = 0$, all vertices belong to a giant component of order 999, except for the isolated vertex 908.

• *random_graph*$(1000,5000,0,0,1,0,0,0,0,0)$ is a digraph with 1000 vertices and 5000 arcs, obtained via a similar sort of evolution. (In fact, each of its arcs happens to be part also of *random_graph*$(1000,5000,0,0,0,0,0,0,0,0)$.)

• *subsets*$(5,1,-10,0,0,0,{}^{\#}\mathtt{1},0)$ is a graph with $\binom{11}{5} = 462$ vertices, one for every five-element subset of $\{0,1,\ldots,10\}$. Two vertices are adjacent whenever the corresponding subsets are disjoint; thus, the graph is regular of degree 6, and it has 1386 edges. We can consider it to be a generalization of the Petersen graph, which has *subsets*$(2,1,-4,0,0,0,{}^{\#}\mathtt{1},0)$ as one of its SGB names.

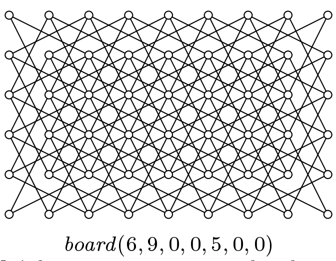
• *subsets*$(5,1,-10,0,0,0,{}^{\#}\mathtt{10},0)$ has the same 462 vertices, but now they are adjacent if the corresponding subsets have four elements in common. This graph is regular of degree 30, and it has 6930 edges.

• *parts*$(30,10,30,0)$ is another SGB graph with a mathematical basis. It has 3590 vertices, one for each partition of 30 into at most 10 parts. Two partitions are adjacent when one is obtained by subdividing a part of the other; this rule defines 31377 edges. The digraph *parts*$(30,10,30,1)$ is similar, but its 31377 arcs point from shorter to longer partitions (for example, `13+7+7+3` $\longrightarrow$ `7+7+7+6+3`).

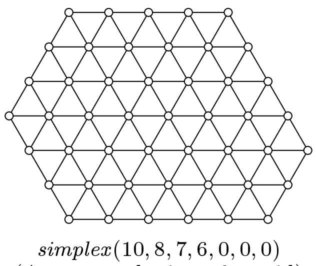
• *simplex*$(10,10,10,10,10,0,0)$ is a graph with 286 vertices and 1320 edges. Its vertices are the integer solutions to $x_1+x_2+x_3+x_4 = 10$ with $x_i \ge 0$, namely the "compositions of 10 into four nonnegative parts"; they can also be regarded as barycentric coordinates for points inside a tetrahedron. The edges, such as `3,1,4,2` — `3,0,4,3`, connect compositions that are as close together as possible.

• *board*$(8,8,0,0,5,0,0)$ and *board*$(8,8,0,0,-2,0,0)$ are graphs on 64 vertices whose 168 or 280 edges correspond to the moves of a knight or bishop in chess.

And zillions of further examples are obtainable by varying the parameters to the SGB graph generators. For example, Fig. 5 shows two simple variants of *board* and *simplex*; the somewhat arcane rules of *board* are explained in exercise 75.



board$(6,9,0,0,5,0,0)$
(Knight moves on a 6×9 chessboard)



simplex$(10,8,7,6,0,0,0)$
(A truncated triangular grid)

Fig. 5. Samples of SGB graphs related to board games.

bipartite
random_graph
pseudorandom
giant component
isolated vertex
random_graph
evolution
digraph, random
random_graph
subsets
disjoint
Petersen graph
parts
partition
simplex
compositions of 10 into four nonnegative parts
barycentric coordinates
tetrahedron
board
knight
bishop
chess
triangular grid

graphs, algebra of++
complement
null graph
adjacency matrices
J
all 1s
I
identity matrix
line graph
$L(G)$
k-edge-colorable
union
rook
bishop
queen
board
gunion
juxtaposition
direct sum
complete k-partite graphs
scandalous fact
notation
complete graphs
disjoint graphs
join
Zykov
directed join

Graph algebra. We can also obtain new graphs by operating on the graphs that we already have. For example, if $G = (V, E)$ is any graph, its *complement* $\overline{G} = (V, \overline{E})$ is obtained by letting

$$u \longrel v \text{ in } \overline{G} \qquad \Longleftrightarrow \qquad u \neq v \text{ and } u \not\longrel v \text{ in } G. \tag{35}$$

Thus, non-edges become edges, and vice versa. Notice that $\overline{\overline{G}} = G$, and that $\overline{K_n}$ has no edges. The corresponding adjacency matrices A and $\overline{A}$ satisfy

$$A + \overline{A} \;=\; J - I; \tag{36}$$

here J is the matrix of all 1s, and I is the identity matrix, so J and $J - I$ are respectively the adjacency matrices of J_n and K_n when G has order n.

Furthermore, every graph $G = (V, E)$ leads to a *line graph* $L(G)$, whose vertices are the edges E; two edges are adjacent in $L(G)$ if they have a common vertex. Thus, for example, the line graph $L(K_n)$ has $\binom{n}{2}$ vertices, and it is regular of degree $2n - 4$ when $n \geq 2$ (see exercise 82). A graph is called *k-edge-colorable* when its line graph is k-colorable.

Given two graphs $G = (U, E)$ and $H = (V, F)$, their *union* $G \cup H$ is the graph $(U \cup V, E \cup F)$ obtained by combining the vertices and edges. For example, suppose G and H are the graphs of rook and bishop moves in chess; then $G \cup H$ is the graph of queen moves, and its official SGB name is

$$\mathit{gunion}(\mathit{board}(8, 8, 0, 0, -1, 0, 0), \mathit{board}(8, 8, 0, 0, -2, 0, 0), 0, 0). \tag{37}$$

In the special case where the vertex sets U and V are disjoint, the union $G \cup H$ doesn't require the vertices to be identified in any consistent way for cross-correlation; we get a diagram for $G \cup H$ by simply drawing a diagram of G next to a diagram of H. This special case is called the "juxtaposition" or *direct sum* of G and H, and we shall denote it by $G \oplus H$. For example, it's easy to see that

$$K_m \oplus K_n \;\cong\; \overline{K_{m,n}}, \tag{38}$$

and that every graph is the direct sum of its connected components.

Equation (38) is a special case of the general formula

$$K_{n_1} \oplus K_{n_2} \oplus \cdots \oplus K_{n_k} \;\cong\; \overline{K_{n_1, n_2, \ldots, n_k}}, \tag{39}$$

which holds for complete k-partite graphs whenever $k \geq 2$. But (39) fails when $k = 1$, because of a scandalous fact: The standard graph-theoretic notation for complete graphs is inconsistent! Indeed, $K_{m,n}$ denotes a complete 2-partite graph, but K_n does *not* denote a complete 1-partite graph. Somehow graph theorists have been able to live with this anomaly for decades without going berserk.

Another important way to combine disjoint graphs G and H is to form their *join*, $G \longrel H$, which consists of $G \oplus H$ together with all edges $u \longrel v$ for $u \in U$ and $v \in V$. [See A. A. Zykov, *Mat. Sbornik* **24** (1949), 163–188, §I.3.] And if G and H are disjoint *digraphs*, their *directed join* $G \longrightarrow H$ is similar, but it supplements $G \oplus H$ by adding only the one-way arcs $u \longrightarrow v$ from U to V.

The direct sum of two matrices A and B is obtained by placing B diagonally below and to the right of A:

$$A \oplus B \;=\; \begin{pmatrix} A & O \\ O & B \end{pmatrix}, \tag{40}$$

where each O in this example is a matrix of all zeros, with the proper number of rows and columns to make everything line up correctly. Our notation $G \oplus H$ for the direct sum of graphs is easy to remember because the adjacency matrix for $G \oplus H$ is precisely the direct sum of the respective adjacency matrices A and B for G and H. Similarly, the adjacency matrices for $G \mathbin{—} H$, $G \longrightarrow H$, and $G \longleftarrow H$ are

$$A \mathbin{—} B = \begin{pmatrix} A & J \\ J & B \end{pmatrix}, \qquad A \longrightarrow B = \begin{pmatrix} A & J \\ O & B \end{pmatrix}, \qquad A \longleftarrow B = \begin{pmatrix} A & O \\ J & B \end{pmatrix}, \tag{41}$$

respectively, where J is an all-1s matrix as in (36). These operations are associative, and related by complementation:

$$A \oplus (B \oplus C) = (A \oplus B) \oplus C, \qquad A \mathbin{—} (B \mathbin{—} C) = (A \mathbin{—} B) \mathbin{—} C; \tag{42}$$

$$A \longrightarrow (B \longrightarrow C) = (A \longrightarrow B) \longrightarrow C, \qquad A \longleftarrow (B \longleftarrow C) = (A \longleftarrow B) \longleftarrow C; \tag{43}$$

$$\overline{A \oplus B} = \overline{A} \mathbin{—} \overline{B}, \qquad \overline{A \mathbin{—} B} = \overline{A} \oplus \overline{B}; \tag{44}$$

$$\overline{A \longrightarrow B} = \overline{A} \longleftarrow \overline{B}, \qquad \overline{A \longleftarrow B} = \overline{A} \longrightarrow \overline{B}; \tag{45}$$

$$(A \oplus B) + (A \mathbin{—} B) \;=\; (A \longrightarrow B) + (A \longleftarrow B). \tag{46}$$

Notice that, by combining (39) with (42) and (44), we have

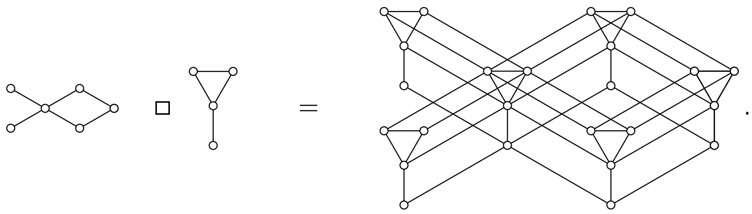
$$K_{n_1,n_2,\ldots,n_k} \;=\; \overline{K_{n_1}} \mathbin{—} \overline{K_{n_2}} \mathbin{—} \cdots \mathbin{—} \overline{K_{n_k}} \tag{47}$$

when $k \ge 2$. Also

$$K_n = K_1 \mathbin{—} K_1 \mathbin{—} \cdots \mathbin{—} K_1 \quad\text{and}\quad K_n^{\rightarrow} = K_1 \longrightarrow K_1 \longrightarrow \cdots \longrightarrow K_1\,, \tag{48}$$

with n copies of K_1, showing that $K_n = K_{1,1,\ldots,1}$ is a complete n-partite graph.

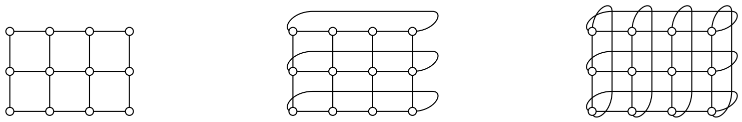
Direct sums and joins are analogous to addition, because we have $\overline{K_m} \oplus \overline{K_n} = \overline{K_{m+n}}$ and $K_m \mathbin{—} K_n = K_{m+n}$. We can also combine graphs with algebraic operations that are analogous to multiplication. For example, the *Cartesian product* operation forms a graph $G \,\square\, H$ of order mn from a graph $G = (U, E)$ of order m and a graph $H = (V, F)$ of order n. The vertices of $G \,\square\, H$ are ordered pairs (u, v), where $u \in U$ and $v \in V$; the edges are $(u, v) \mathbin{—} (u', v)$ when $u \mathbin{—} u'$ in G, together with $(u, v) \mathbin{—} (u, v')$ when $v \mathbin{—} v'$ in H. In other words, $G \,\square\, H$ is formed by replacing each vertex of G by a copy of H, and replacing each edge of G by edges between corresponding vertices of the appropriate copies:



(49)

direct sum of two matrices
O
direct sum of graphs
adjacency matrix
J
associative
complementation
transitive tournament
$K_n^{\rightarrow}$
K_n
Cartesian product

As usual, the simplest special cases of this general construction turn out to be especially important in practice. When both G and H are paths or cycles, we get "graph-paper graphs," namely the $m \times n$ *grid* $P_m \square P_n$, the $m \times n$ *cylinder* $P_m \square C_n$, and the $m \times n$ *torus* $C_m \square C_n$, illustrated here for $m = 3$ and $n = 4$:



(50)

$P_3 \square P_4$ (3 × 4 grid) $P_3 \square C_4$ (3 × 4 cylinder) $C_3 \square C_4$ (3 × 4 torus)

Four other noteworthy ways to define products of graphs have also proved to be useful. In each case the vertices of the product graph are ordered pairs (u, v).

- The *direct product* $G \otimes H$, also called the "conjunction" of G and H, or their "categorical product," has $(u, v) — (u', v')$ when $u — u'$ in G and $v — v'$ in H.
- The *strong product* $G \boxtimes H$ combines the edges of $G \square H$ with those of $G \otimes H$.
- The *odd product* $G \triangle H$ has $(u, v) — (u', v')$ when we have either $u — u'$ in G or $v — v'$ in H, but not both.
- The *lexicographic product* $G \circ H$, also called the "composition" of G and H, has $(u, v) — (u', v')$ when $u — u'$ in G, and $(u, v) — (u, v')$ when $v — v'$ in H.

All five of these operations extend naturally to products of $k \ge 2$ graphs $G_1 = (V_1, E_1)$, ..., $G_k = (V_k, E_k)$, whose vertices are the ordered k-tuples $(v_1, \ldots, v_k)$ with $v_j \in V_j$ for $1 \le j \le k$. For example, when $k = 3$, the Cartesian products $G_1 \square (G_2 \square G_3)$ and $(G_1 \square G_2) \square G_3$ are isomorphic, if we consider the compound vertices $(v_1, (v_2, v_3))$ and $((v_1, v_2), v_3)$ to be the same as (v_1, v_2, v_3). Therefore we can write this Cartesian product without parentheses, as $G_1 \square G_2 \square G_3$. The most important example of a Cartesian product with k factors is the k-cube,

$$K_2 \square K_2 \square \cdots \square K_2; \tag{51}$$

its 2^k vertices $(v_1, \ldots, v_k)$ are adjacent when their Hamming distance is 1.

In general, suppose $v = (v_1, \ldots, v_k)$ and $v' = (v'_1, \ldots, v'_k)$ are k-tuples of vertices, where we have $v_j — v'_j$ in G_j for exactly a of the subscripts j, and $v_j = v'_j$ for exactly b of the subscripts. Then we have:

- $v — v'$ in $G_1 \square \cdots \square G_k$ if and only if $a = 1$ and $b = k - 1$;
- $v — v'$ in $G_1 \otimes \cdots \otimes G_k$ if and only if $a = k$ and $b = 0$;
- $v — v'$ in $G_1 \boxtimes \cdots \boxtimes G_k$ if and only if $a + b = k$ and $a > 0$;
- $v — v'$ in $G_1 \triangle \cdots \triangle G_k$ if and only if a is odd.

The lexicographic product is somewhat different, because it isn't commutative; in $G_1 \circ \cdots \circ G_k$ we have $v — v'$ for $v \ne v'$ if and only if $v_j — v'_j$, where j is the minimum subscript with $v_j \ne v'_j$.

Exercises 91–102 explore some of the basic properties of graph products. See also the book *Product Graphs* by Wilfried Imrich and Sandi Klavžar (2000), which contains a comprehensive introduction to the general theory, including algorithms for factorization of a given graph into "prime" subgraphs.

graph-paper graphs
grid
cylinder
torus
direct product
conjunction
categorical product
tensor product, see direct product
strong product
odd product
lexicographic product
composition
associative laws
k-cube
Hamming distance
Imrich
Klavžar
factorization of a given graph
"prime" subgraphs

***Graphical degree sequences.** A sequence $d_1 d_2 \ldots d_n$ of nonnegative integers is called *graphical* if there's at least one graph on vertices $\{1, 2, \ldots, n\}$ such that vertex k has degree d_k. We can assume that $d_1 \geq d_2 \geq \cdots \geq d_n$. Clearly $d_1 < n$ in any such graph; and the sum $m = d_1 + d_2 + \cdots + d_n$ of any graphical sequence is always even, because it is twice the number of edges. Furthermore, it's easy to see that the sequence 3311 is not graphical; therefore graphical sequences must also satisfy additional conditions. What are they?

degree sequences++
graphical
Havel (mathematician)
tableaux+
conjugate partitions+

A simple way to decide if a given sequence $d_1 d_2 \ldots d_n$ is graphical, and to construct such a graph if one exists, was discovered by V. Havel [*Časopis pro Pěstování Matematiky* **80** (1955), 477–479]. We begin with an empty tableau, having d_k cells in row k; these cells represent "slots" into which we'll place the neighbors of vertex k in the constructed graph. Let c_j be the number of cells in column j; thus $c_1 \geq c_2 \geq \cdots$, and when $1 \leq k \leq n$ we have $c_j \geq k$ if and only if $d_k \geq j$. For example, suppose $n = 8$ and $d_1 \ldots d_8 = 55544322$; then

1					
2					
3					
4					
5					
6					
7					
8					

(52)

is the initial tableau, and we have $c_1 \ldots c_5 = 88653$. Havel's idea is to pair up vertex n with d_n of the highest-degree vertices. In this case, for example, we create the two edges $8 — 3$ and $8 — 2$, and the tableau takes the following form:

1					
2					8
3					8
4					
5					
6					
7					
8	2	3			

. (53)

(We don't want $8 — 1$, because the empty slots should continue to form a tableau shape; the cells of each column must be filled from the bottom up.) Next we set $n \leftarrow 7$ and create two further edges, $7 — 1$ and $7 — 5$. And then come three more, $6 — 4$, $6 — 3$, $6 — 2$, making the tableau almost half full:

1					7
2				6	8
3				6	8
4				6	
5				7	
6	2	3	4		
7	5	1			
8	2	3			

. (54)

majorization
partitions

We've reduced the problem to finding a graph with degree sequence $d_1 \dots d_5 = 43333$; at this point we also have $c_1 \dots c_4 = 5551$. The reader is encouraged to fill in the remaining blanks, before looking at the answer in exercise 103.

Algorithm H (*Graph generator for specified degrees*). Given $d_1 \geq \cdots \geq d_n \geq d_{n+1} = 0$, this algorithm creates edges between the vertices $\{1, \dots, n\}$ in such a way that exactly d_k edges touch vertex k, for $1 \leq k \leq n$, unless the sequence $d_1 \dots d_n$ isn't graphical. An array $c_1 \dots c_{d_1}$ is used for auxiliary storage.

H1. [Set the c's.] Start with $k \leftarrow d_1$ and $j \leftarrow 0$. Then while $k > 0$ do the following operations: Set $j \leftarrow j + 1$; while $k > d_{j+1}$, set $c_k \leftarrow j$ and $k \leftarrow k - 1$.

H2. [Find n.] Set $n \leftarrow c_1$. Terminate successfully if $n = 0$; terminate unsuccessfully if $d_1 \geq n > 0$.

H3. [Loop on j.] Set $i \leftarrow 1$, $t \leftarrow d_1$, and $r \leftarrow c_t$. Do step H4 for $j = d_n$, $d_n - 1$, $\dots$, 1; then return to H2.

H4. [Generate a new edge.] Set $c_j \leftarrow c_j - 1$ and $k \leftarrow c_t$. Create the edge $k \longrightarrow n$, and set $d_k \leftarrow d_k - 1$, $c_t \leftarrow k - 1$. If $k = i$, set $i \leftarrow r + 1$, $t \leftarrow d_i$, and $r \leftarrow c_t$. (See exercise 104.) ▮

When Algorithm H succeeds, it certainly has constructed a graph with the desired degrees. But when it fails, how can we be sure that its mission was impossible? The key fact is based on an important concept called "majorization": If $d_1 \dots d_n$ and $d'_1 \dots d'_n$ are two partitions of the same integer (that is, if $d_1 \geq \cdots \geq d_n$ and $d'_1 \geq \cdots \geq d'_n$ and $d_1 + \cdots + d_n = d'_1 + \cdots + d'_n$), we say that $d_1 \dots d_n$ *majorizes* $d'_1 \dots d'_n$ if $d_1 + \cdots + d_k \geq d'_1 + \cdots + d'_k$ for $1 \leq k \leq n$.

Lemma M. *If $d_1 \dots d_n$ is graphical and $d_1 \dots d_n$ majorizes $d'_1 \dots d'_n$, then $d'_1 \dots d'_n$ is also graphical.*

Proof. It is sufficient to prove the claim when $d_1 \dots d_n$ and $d'_1 \dots d'_n$ differ in only two places,

$$d'_k = d_k - [k = i] + [k = j] \qquad \text{where } i < j, \tag{55}$$

because any sequence majorized by $d_1 \dots d_n$ can be obtained by repeatedly performing mini-majorizations such as this. (Exercise 7.2.1.4–55 discusses majorization in detail.)

Condition (55) implies that $d_i > d'_i \geq d'_{i+1} \geq d'_j > d_j$. So any graph with degree sequence $d_1 \dots d_n$ contains a vertex v such that $v \longrightarrow i$ and $v \not\longrightarrow j$. Deleting the edge $v \longrightarrow i$ and adding the edge $v \longrightarrow j$ yields a graph with degree sequence $d'_1 \dots d'_n$, as desired. ▮

Corollary H. *Algorithm H succeeds whenever $d_1 \dots d_n$ is graphical.*

Proof. We may assume that $n > 1$. Suppose G is any graph on $\{1, \dots, n\}$ with degree sequence $d_1 \dots d_n$, and let G' be the subgraph induced by $\{1, \dots, n-1\}$; in other words, obtain G' by removing vertex n and the d_n edges that it touches. The degree sequence $d'_1 \dots d'_{n-1}$ of G' is obtained from $d_1 \dots d_{n-1}$ by reducing some d_n of the entries by 1 and sorting them into nonincreasing order. By definition, $d'_1 \ldots d'_{n-1}$ is graphical. The new degree sequence $d''_1 \ldots d''_{n-1}$ produced by the strategy of steps H3 and H4 is designed to be majorized by every such $d'_1 \ldots d'_{n-1}$, because it reduces the largest possible d_n entries by 1. Thus the new $d''_1 \ldots d''_{n-1}$ is graphical. Algorithm H, which sets $d_1 \ldots d_{n-1} \leftarrow d''_1 \ldots d''_{n-1}$, will therefore succeed by induction on n. ▮

The running time of Algorithm H is roughly proportional to the number of edges generated, which can be of order n^2. Exercise 105 presents a faster method, which decides in $O(n)$ steps whether or not a given sequence $d_1 \ldots d_n$ is graphical (without constructing any graph).

Beyond graphs. When the vertices and/or arcs of a graph or digraph are decorated with additional data, we call it a *network*. For example, every vertex of *words*$(5757, 0, 0, 0)$ has an associated rank, which corresponds to the popularity of the corresponding five-letter word. Every vertex of *plane_lisa*$(360, 250, 15, 0, 360, 0, 250, 0, 0, 2295000)$ has an associated pixel density, between 0 and 15. Every arc of *board*$(8, 8, 0, 0, -2, 0, 0)$ has an associated length, which reflects the distance of a piece's motion on the board: A bishop's move from corner to corner has length 7. The Stanford GraphBase includes several further generators that were not mentioned above, because they are primarily used to generate interesting networks, rather than to generate graphs with interesting structure:

• *miles*$(128, 0, 0, 0, 0, 127, 0)$ is a network with 128 vertices, corresponding to the same North American cities as the graph *plane_miles* described earlier. But *miles*, unlike *plane_miles*, is a complete graph with $\binom{128}{2}$ edges. Every edge has an integer length, which represents the distance that a car or truck would have needed to travel in 1949 when going from one given city to another. For example, '`Vancouver, BC`' is 3496 miles from '`West Palm Beach, FL`' in the *miles* network.

• *econ*$(81, 0, 0, 0)$ is a network with 81 vertices and 4902 arcs. Its vertices represent sectors of the United States economy, and its arcs represent the flow of money from one sector to another during the year 1985, measured in millions of dollars. For example, the flow value from `Apparel` to `Household furniture` is 44, meaning that the furniture industry paid $44,000,000 to the apparel industry in that year. The sum of flows coming into each vertex is equal to the sum of flows going out. An arc appears only when the flow is nonzero. A special vertex called `Users` receives the flows that represent total demand for a product; a few of these end-user flows are negative, because of the way imported goods are treated by government economists.

• *games*$(120, 0, 0, 0, 0, 0, 128, 0)$ is a network with 120 vertices and 1276 arcs. Its vertices represent football teams at American colleges and universities. Arcs run between teams that played each other during the exciting 1990 season, and they are labeled with the number of points scored. For example, the arc `Stanford` $\longrightarrow$ `California` has value 27, and the arc `California` $\longrightarrow$ `Stanford` has value 25, because the Stanford Cardinal defeated the U. C. Berkeley Golden Bears by a score of 27–25 on 17 November 1990.

• *risc*(16) is a network of an entirely different kind. It has 3240 vertices and 7878 arcs, which define a *directed acyclic graph* or "dag" — namely, a digraph

network
words
plane_lisa
pixel
board
bishop
Stanford GraphBase
miles
plane_miles
highways
econ
flow of money
games
football
colleges
universities
Stanford Cardinal
Golden Bears
risc
directed acyclic graph
dag

that contains no oriented cycles. The vertices represent gates that have Boolean values; an arc such as `Z45` $\longrightarrow$ `R0:7~` means that the value of gate `Z45` is an input to gate `R0:7~`. Each gate has a type code (AND, OR, XOR, NOT, latch, or external input); each arc has a length, denoting an amount of delay. The network contains the complete logic for a miniature RISC chip that is able to obey simple commands governing sixteen registers, each 16 bits wide.

Complete details about all the SGB generators can be found in the author's book *The Stanford GraphBase* (New York: ACM Press, 1993), together with dozens of short example programs that explain how to manipulate the graphs and networks that the generators produce. For example, a program called LADDERS shows how to find a shortest path between one five-letter word and another. A program called TAKE_RISC demonstrates how to put a nanocomputer through its paces by simulating the actions of a network built from the gates of *risc*(16).

Hypergraphs. Graphs and networks can be utterly fascinating, but they aren't the end of the story by any means. Lots of important combinatorial algorithms are designed to work with *hypergraphs*, which are more general than graphs because their edges are allowed to be *arbitrary* subsets of the vertices.

For example, we might have seven vertices, identified by nonzero binary strings $v = a_1a_2a_3$, together with seven edges, identified by bracketed nonzero binary strings $e = [b_1b_2b_3]$, with $v \in e$ if and only if $(a_1b_1+a_2b_2+a_3b_3) \bmod 2 = 0$. Each of these edges contains exactly three vertices:

$$\begin{aligned}&[001] = \{010, 100, 110\}; \quad [010] = \{001, 100, 101\}; \quad [011] = \{011, 100, 111\};\\ &\qquad [100] = \{001, 010, 011\}; \quad [101] = \{010, 101, 111\};\\ &\qquad [110] = \{001, 110, 111\}; \quad [111] = \{011, 101, 110\}.\end{aligned} \tag{56}$$

And by symmetry, each vertex belongs to exactly three edges. (Edges that contain three or more vertices are sometimes called "hyperedges," to distinguish them from the edges of an ordinary graph. But it's OK to call them just "edges.")

A hypergraph is said to be *r-uniform* if every edge contains exactly r vertices. Thus (56) is a 3-uniform hypergraph, and a 2-uniform hypergraph is an ordinary graph. The complete r-uniform hypergraph $K_n^{(r)}$ has n vertices and $\binom{n}{r}$ edges.

Most of the basic concepts of graph theory can be extended to hypergraphs in a natural way. For example, if $H = (V, E)$ is a hypergraph and if $U \subseteq V$, the subhypergraph $H \mid U$ induced by U has the edges $\{e \mid e \in E \text{ and } e \subseteq U\}$. The complement $\overline{H}$ of an r-uniform hypergraph has the edges of $K_n^{(r)}$ that aren't edges of H. A k-coloring of a hypergraph is an assignment of colors to the vertices so that no edge is monochromatic. And so on.

Hypergraphs go by many other names, because the same properties can be formulated in many different ways. For example, every hypergraph $H = (V, E)$ is essentially a *family of sets*, because each edge is a subset of V. A 3-uniform hypergraph is also called a *triple system*. A hypergraph is also equivalent to a matrix B of 0s and 1s, with one row for each vertex v and one column for each edge e; row v and column e of this matrix contains the value $b_{ve} = [v \in e]$.

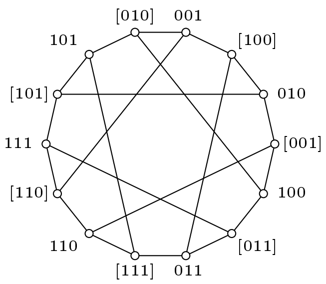
oriented cycles
Acyclic: Containing no cycles
RISC: Reduced Instruction Set Computer
gates
Boolean values
AND
OR
XOR
NOT
latch
Knuth
Stanford GraphBase: complete guide to
LADDERS
shortest path
TAKE_RISC
nanocomputer
risc
hypergraphs+++
hyperedges
edges of a hypergraph+++
r-uniform
complete r-uniform hypergraph
$K_n^{(r)}$
induced
complement $\overline{H}$ of an r-uniform hypergraph
k-coloring of a hypergraph
family of sets
set system, see hypergraph
triple system
0–1 matrices++

incidence matrix
bipartite graph
connected
cycle
dual
transposing
regular
distinct columns
Boolean function

Matrix B is called the *incidence matrix* of H, and we say that "v is incident with e" when $v \in e$. Furthermore, a hypergraph is equivalent to a *bipartite graph*, with vertex set $V \cup E$ and with the edge v — e whenever v is incident with e. The hypergraph is said to be *connected* if and only if the corresponding bipartite graph is connected. A *cycle* of length k in a hypergraph is defined to be a cycle of length $2k$ in the corresponding bipartite graph.

For example, the hypergraph (56) can be defined by an equivalent incidence matrix or an equivalent bipartite graph as follows:

$$\begin{array}{c} \\ \scriptstyle 001 \\ \scriptstyle 010 \\ \scriptstyle 011 \\ \scriptstyle 100 \\ \scriptstyle 101 \\ \scriptstyle 110 \\ \scriptstyle 111 \end{array}
\begin{array}{c} \begin{array}{ccccccc} \scriptstyle [001] & \scriptstyle [010] & \scriptstyle [011] & \scriptstyle [100] & \scriptstyle [101] & \scriptstyle [110] & \scriptstyle [111] \end{array} \\
\begin{pmatrix} 0 & 1 & 0 & 1 & 0 & 1 & 0 \\ 1 & 0 & 0 & 1 & 1 & 0 & 0 \\ 0 & 0 & 1 & 1 & 0 & 0 & 1 \\ 1 & 1 & 1 & 0 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 & 1 & 0 & 1 \\ 1 & 0 & 0 & 0 & 0 & 1 & 1 \\ 0 & 0 & 1 & 0 & 1 & 1 & 0 \end{pmatrix} \end{array}$$



(57)

It contains 28 cycles of length 3, such as

$$[101] \text{ — } 101 \text{ — } [010] \text{ — } 001 \text{ — } [100] \text{ — } 010 \text{ — } [101]. \tag{58}$$

The *dual* H^T of a hypergraph H is obtained by interchanging the roles of vertices and edges, but retaining the incidence relation. In other words, it corresponds to transposing the incidence matrix. Notice, for example, that the dual of an r-regular graph is an r-uniform hypergraph.

Incidence matrices and bipartite graphs might correspond to hypergraphs in which some edges occur more than once, because distinct columns of the matrix might be equal. When a hypergraph $H = (V, E)$ does not have any repeated edges, it corresponds also to yet another combinatorial object, namely a *Boolean function*. For if, say, the vertex set V is $\{1, 2, \ldots, n\}$, the function

$$h(x_1, x_2, \ldots, x_n) \;=\; \bigl[\{j \mid x_j = 1\} \in E\bigr] \tag{59}$$

characterizes the edges of H. For example, the Boolean formula

$$\begin{aligned} &(x_1 \oplus x_2 \oplus x_4) \wedge (x_2 \oplus x_3 \oplus x_5) \wedge (x_3 \oplus x_4 \oplus x_6) \wedge (x_4 \oplus x_5 \oplus x_7) \\ &\qquad \wedge (x_5 \oplus x_6 \oplus x_1) \wedge (x_6 \oplus x_7 \oplus x_2) \wedge (x_7 \oplus x_1 \oplus x_3) \wedge (\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3) \end{aligned} \tag{60}$$

is another way to describe the hypergraph of (56) and (57).

The fact that combinatorial objects can be viewed in so many ways can be mind-boggling. But it's also extremely helpful, because it suggests different ways to solve equivalent problems. When we look at a problem from different perspectives, our brains naturally think of different ways to attack it. Sometimes we get the best insights by thinking about how to manipulate rows and columns in a matrix. Sometimes we make progress by imagining vertices and paths, or by visualizing clusters of points in space. Sometimes Boolean algebra is just the thing. If we're stuck in one domain, another might come to our rescue.

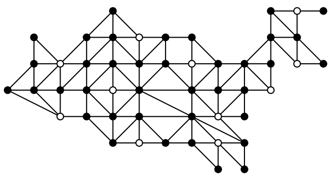
Covering and independence. If $H = (V, E)$ is a graph or hypergraph, a set U of vertices is said to *cover* H if every edge contains at least one member of U. A set W of vertices is said to be *independent* (or "stable") in H if no edge is completely contained in W.

From the standpoint of the incidence matrix, a covering is a set of rows whose sum is nonzero in every column. And in the special case that H is a graph, every column of the matrix contains just two 1s; hence an independent set in a graph corresponds to a set of rows that are mutually orthogonal — that is, a set for which the dot product of any two different rows is zero.

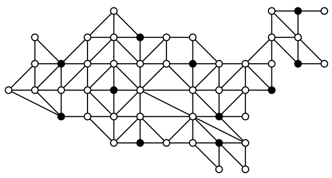
These concepts are opposite sides of the same coin. If U covers H, then $W = V \setminus U$ is independent in H; conversely, if W is independent in H, then $U = V \setminus W$ covers H. Both statements are equivalent to saying that the induced hypergraph $H \mid W$ has no edges.

This dual relationship between covering and independence, which was perhaps first noted by Claude Berge [*Proc. National Acad. Sci.* **43** (1957), 842–844], is somewhat paradoxical. Although it's logically obvious and easy to verify, it's also intuitively surprising. When we look at a graph and try to find a large independent set, we tend to have rather different thoughts from when we look at the same graph and try to find a small vertex cover; yet both goals are the same.

A covering set U is *minimal* if $U \setminus u$ fails to be a cover for all $u \in U$. Similarly, an independent set W is *maximal* if $W \cup w$ fails to be independent for all $w \notin W$. Here, for example, is a minimal cover of the 49-vertex graph of the contiguous United States, (17), and the corresponding maximal independent set:



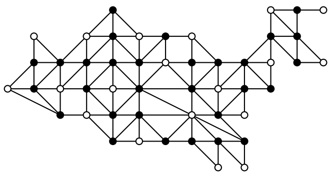
Minimal vertex cover, with 38 vertices



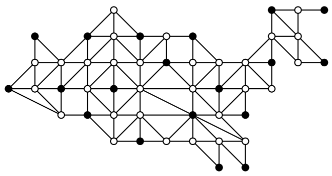
Maximal independent set, with 11 vertices

(61)

A covering is called *minimum* if it has the smallest possible size, and an independent set is called *maximum* if it has the largest possible size. For example, with graph (17) we can do much better than (61):



Minimum vertex cover, with 30 vertices



Maximum independent set, with 19 vertices

(62)

Notice the subtle distinction between "minimal" and "minimum" here: In general (but in contrast to most dictionaries of English), people who work with combinatorial algorithms use '-al' words like "minimal" or "optimal" to refer

cover
independent
stable
orthogonal
dot product
Berge
minimal
maximal
contiguous United States
dictionaries of English
minimal versus minimum+
maximal versus maximum+
optimal versus optimum

to combinatorial configurations that are *locally* best, in the sense that small changes don't improve them. The corresponding '-um' words, "minimum" or "optimum," are reserved for configurations that are *globally* best, considered over all possibilities. It's easy to find solutions to any optimization problem that are merely optimal, in the weak local sense, by climbing repeatedly until reaching the top of a hill. But it's usually much harder to find solutions that are truly optimum. For example, we'll see in Section 7.9 that the problem of finding a maximum independent set in a given graph belongs to a class of difficult problems that are called *NP-complete.*

Even when a problem is NP-complete, we needn't despair. We'll discuss techniques for finding minimum covers in several parts of this chapter, and those methods work fine on smallish problems; the optimum solution in (62) was found in less than a second, after examining only a tiny fraction of the 2^{49} possibilities. Furthermore, special cases of NP-complete problems often turn out to be simpler than the general case. In Section 7.5.1 we'll see that a minimum vertex cover can be discovered quickly in any bipartite graph, or in any hypergraph that is the dual of a graph. And in Section 7.5.5 we'll study efficient ways to discover a maximum *matching*, which is a maximum independent set in the line graph of a given graph.

The problem of maximizing the size of an independent set occurs sufficiently often that it has acquired a special notation: If H is any hypergraph, the number

$$\alpha(H) \;=\; \max\bigl\{|W| \bigm| W \text{ is an independent set of vertices in } H\bigr\} \qquad (63)$$

is called the *independence number* (or the stability number) of H. Similarly,

$$\chi(H) \;=\; \min\{k \mid H \text{ is } k\text{-colorable}\} \qquad (64)$$

is called the *chromatic number* of H. Notice that $\chi(H)$ is the size of a minimum covering of H by independent sets, because the vertices that receive any particular color must be independent according to our definitions.

These definitions of $\alpha(H)$ and $\chi(H)$ apply in particular to the case when H is an ordinary graph, but of course we usually write $\alpha(G)$ and $\chi(G)$ in such situations. Graphs have another important number called their *clique number*,

$$\omega(G) \;=\; \max\bigl\{|X| \bigm| X \text{ is a clique in } G\bigr\}, \qquad (65)$$

where a "clique" is a set of mutually adjacent vertices. Clearly

$$\omega(G) \;=\; \alpha(\overline{G}), \qquad (66)$$

because a clique in G is an independent set in the complementary graph. Similarly we can see that $\chi(\overline{G})$ is the minimum size of a "clique cover," which is a set of cliques that exactly covers all of the vertices.

Several instances of "exact cover problems" were mentioned earlier in this section, without an explanation of exactly what such a problem really signifies. Finally we're ready for the definition: Given the incidence matrix of a hypergraph H, an *exact cover* of H is a set of rows whose sum is $(1\,1\,\ldots\,1)$. In other words, an exact cover is a set of vertices that touches each hyperedge exactly once; an ordinary cover is only required to touch each hyperedge *at least* once.

locally
globally
maximum independent set
NP-complete
bipartite graph
dual
matching
line graph
$\alpha(H)$
independence number
stability number
chromatic number
clique number
clique
independent set
complementary graph
clique cover
cover
incidence matrix
exact cover
–graphs

EXERCISES

1. [*25*] Suppose $n = 4m - 1$. Construct arrangements of Langford pairs for the numbers $\{1, 1, \ldots, n, n\}$, with the property that we also obtain a solution for $n = 4m$ by changing the first '$2m{-}1$' to '$4m$' and appending '$2m{-}1\ 4m$' at the right. *Hint:* Put the $m - 1$ even numbers $4m{-}4$, $4m{-}6$, …, $2m$ at the left.

2. [*18*] For which n can $\{0, 0, 1, 1, \ldots, n{-}1, n{-}1\}$ be arranged as Langford pairs?

3. [*22*] Suppose we arrange the numbers $\{0, 0, 1, 1, \ldots, n{-}1, n{-}1\}$ in a *circle*, instead of a straight line, with distance k between the two k's. Do we get solutions that are essentially distinct from those of exercise 2?

4. [*M20*] (T. Skolem, 1957.) Show that the Fibonacci string $S_\infty = babbababbabba\ldots$ of exercise 1.2.8–36 leads directly to an infinite sequence $0012132453674\ldots$ of Langford pairs for the set of *all* nonnegative integers, if we simply replace the a's and b's independently by 0, 1, 2, etc., from left to right.

▶ **5.** [*HM22*] If a permutation of $\{1, 1, 2, 2, \ldots, n, n\}$ is chosen at random, what is the probability that the two k's are exactly k positions apart, given k? Use this formula to guess the size of the Langford numbers L_n in (1).

▶ **6.** [*M28*] (M. Godfrey, 2002.) Let $f(x_1, \ldots, x_{2n}) = \prod_{k=1}^{n} \bigl(x_k x_{n+k} \sum_{j=1}^{2n-k-1} x_j x_{j+k+1}\bigr)$.

a) Prove that $\sum_{x_1,\ldots,x_{2n} \in \{-1,+1\}} f(x_1, \ldots, x_{2n}) = 2^{2n+1} L_n$.

b) Explain how to evaluate this sum in $O(4^n n)$ steps. How many bits of precision are needed for the arithmetic?

c) Gain a factor of eight by exploiting the identities

$$f(x_1, \ldots, x_{2n}) = f(-x_1, \ldots, -x_{2n}) = f(x_{2n}, \ldots, x_1) = f(x_1, -x_2, \ldots, x_{2n-1}, -x_{2n}).$$

7. [*M22*] Prove that every Langford pairing of $\{1, 1, \ldots, 16, 16\}$ must have seven uncompleted pairs at some point, when read from left to right.

8. [*23*] The simplest Langford sequence is not only well-balanced; it's *planar*, in the sense that its pairs can be connected up without crossing lines as in (2):

$$2\ 3\ 1\ 2\ 1\ 3\,.$$

Find all of the planar Langford pairings for which $n \le 8$.

9. [*24*] (*Langford triples.*) In how many ways can $\{1, 1, 1, 2, 2, 2, \ldots, 9, 9, 9\}$ be arranged in a row so that consecutive k's are k apart, for $1 \le k \le 9$?

10. [*M20*] Explain how to construct a *magic square* directly from Fig. 1. (Convert each card into a number between 1 and 16, in such a way that the rows, columns, and main diagonals all sum to 34.)

11. [*20*] Extend (5) to a "Hebraic-Græco-Latin" square by appending one of the letters {א, ב, ג, ד} to the two-letter string in each compartment. No letter pair (Latin, Greek), (Latin, Hebrew), or (Greek, Hebrew) should appear in more than one place.

▶ **12.** [*M21*] (L. Euler.) Let $L_{ij} = (i{+}j) \bmod n$ for $0 \le i, j < n$ be the addition table for integers mod n. Prove that a latin square orthogonal to L exists if and only if n is odd.

13. [*M25*] A 10×10 square can be divided into four quarters of size 5×5. A 10×10 latin square formed from the digits $\{0, 1, \ldots, 9\}$ has k "intruders" if its upper left quarter has exactly k elements ≥ 5. (See exercise 14(e) for an example with $k = 3$.) Prove that the square has no orthogonal mate unless there are at least three intruders.

Langford pairs
Skolem
Fibonacci string
Godfrey
well-balanced
planar
Langford triples
magic square
Græco-Latin
Hebrew
Euler
addition table
intruders

14. [*29*] Find all orthogonal mates of the following latin squares:

(a)	(b)	(c)	(d)	(e)
3145926870	2718459036	0572164938	1680397425	7823456019
2819763504	0287135649	6051298473	8346512097	8234067195
9452307168	7524093168	4867039215	9805761342	2340178956
6208451793	1435962780	1439807652	2754689130	3401289567
8364095217;	6390718425;	8324756091;	0538976214;	4012395678.
5981274036	4069271853	7203941586	4963820571	5678912340
4627530981	3102684597	5610473829	7192034658	6789523401
0576148329	9871546302	9148625307	6219405783	0195634782
1730689452	8956307214	2795380164	3471258906	1956740823
7093812645	5643820971	3986512740	5027143869	9567801234

15. [*50*] Find three 10×10 latin squares that are mutually orthogonal to each other.

16. [*48*] (H. J. Ryser, 1967.) A latin square is said to be of "order n" if it has n rows, n columns, and n symbols. Does every latin square of odd order have a transversal?

17. [*25*] Let L be a latin square with elements L_{ij} for $0 \le i, j < n$. Show that the problems of (a) finding all the transversals of L, and (b) finding all the orthogonal mates of L, are special cases of the general exact cover problem.

18. [*M23*] The string $x_1 x_2 \ldots x_N$ is called "n-ary" if each element x_j belongs to the set $\{0, 1, \ldots, n-1\}$ of n-ary digits. Two strings $x_1 x_2 \ldots x_N$ and $y_1 y_2 \ldots y_N$ are said to be *orthogonal* if the N pairs (x_j, y_j) are distinct for $1 \le j \le N$. (Consequently, two n-ary strings cannot be orthogonal if their length N exceeds n^2.) An n-ary matrix with m rows and n^2 columns whose rows are orthogonal to each other is called an *orthogonal array* of order n and depth m.

Find a correspondence between orthogonal arrays of depth m and lists of $m - 2$ mutually orthogonal latin squares. What orthogonal array corresponds to exercise 11?

▸ **19.** [*M25*] Continuing exercise 18, prove that an orthogonal array of order $n > 1$ and depth m is possible only if $m \le n + 1$. Show that this upper limit is achievable when n is a prime number p. Write out an example when $p = 5$.

20. [*HM20*] Show that if each element k in an orthogonal array is replaced by $e^{2\pi k i/n}$, the rows become orthogonal vectors in the usual sense (their dot product is zero).

▸ **21.** [*M21*] A *geometric net* is a system of points and lines that obeys three axioms:

i) Each line is a set of points.

ii) Distinct lines have at most one point in common.

iii) If p is a point and L is a line with $p \notin L$, then there is exactly one line M such that $p \in M$ and $L \cap M = \emptyset$.

If $L \cap M = \emptyset$ we say that L is *parallel* to M, and write $L \parallel M$.

a) Prove that the lines of a geometric net can be partitioned into equivalence classes, with two lines in the same class if and only if they are equal or parallel.

b) Show that if there are at least two classes of parallel lines, every line contains the same number of points as the other lines in its class.

c) Furthermore, if there are at least three classes, there are numbers m and n such that all points belong to exactly m lines and all lines contain exactly n points.

▸ **22.** [*M22*] Show that every orthogonal array can be regarded as a geometric net. Is the converse also true?

23. [*M21*] (*Error-correcting codes.*) The "Hamming distance" $d(x, y)$ between two strings $x = x_1 \ldots x_N$ and $y = y_1 \ldots y_N$ is the number of positions j where $x_j \ne y_j$. A

mutually orthogonal
Ryser
order n
Order of a latin square
latin square
transversals
orthogonal mates
exact cover problem
n-ary
orthogonal
orthogonal array
orthogonal vectors
dot product
geometric net
parallel
orthogonal array
geometric net
Error-correcting codes+
Hamming distance+

b-ary code with n information digits and r check digits is a set $C(b,n,r)$ of b^n strings $x = x_1 \ldots x_{n+r}$, where $0 \le x_j < b$ for $1 \le j \le n+r$. When a codeword x is transmitted and the message y is received, $d(x,y)$ is the number of transmission errors. The code is called *t-error correcting* if we can reconstruct the value of x whenever a message y is received with $d(x,y) \le t$. The *distance* of the code is the minimum value of $d(x,x')$, taken over all pairs of codewords $x \ne x'$.

a) Prove that a code is t-error correcting if and only if its distance exceeds $2t$.
b) Prove that a single-error correcting b-ary code with 2 information digits and 2 check digits is equivalent to a pair of orthogonal latin squares of order b.
c) Furthermore, a code $C(b,2,r)$ with distance $r+1$ is equivalent to a set of r mutually orthogonal latin squares of order b.

▶ **24.** [*M30*] A geometric net with N points and R lines leads naturally to the binary code $C(2,N,R)$ with codewords $x_1 \ldots x_N x_{N+1} \ldots x_{N+R}$ defined by the parity bits

$$x_{N+k} = f_k(x_1,\ldots,x_N) = \bigl(\textstyle\sum\{x_j \mid \text{point } j \text{ lies on line } k\}\bigr) \bmod 2.$$

a) If the net has m classes of parallel lines, prove that this code has distance $m+1$.
b) Find an efficient way to correct up to t errors with this code, assuming that $m = 2t$. Illustrate the decoding process in the case $N = 25$, $R = 30$, $t = 3$.

25. [*27*] Find a latin square whose rows and columns are five-letter words. (For this exercise you'll need to dig out the big dictionaries.)

▶ **26.** [*25*] Compose a meaningful English sentence that contains only five-letter words.

27. [*20*] How many SGB words contain exactly k distinct letters, for $1 \le k \le 5$?

28. [*20*] Are there any pairs of SGB word vectors that differ by ± 1 in each component?

29. [*20*] Find all SGB words that are *palindromes* (equal to their reflection), or mirror pairs (like `regal lager`).

▶ **30.** [*20*] The letters of `first` are in alphabetic order from left to right. What is the lexicographically *first* such five-letter word? What is the last?

31. [*21*] (C. McManus.) Find all sets of three SGB words that are in arithmetic progression but have no common letters in any fixed position. (One such example is $\{$`power`, `slugs`, `visit`$\}$.)

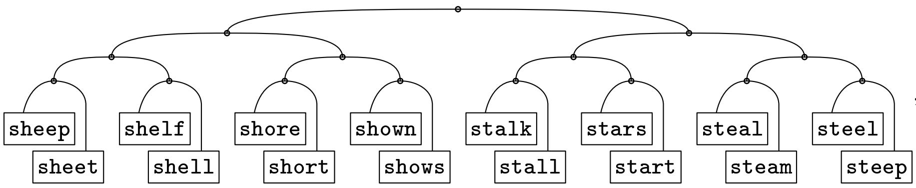
32. [*23*] Does the English language contain any 10-letter words $a_0a_1\ldots a_9$ for which both $a_0a_2a_4a_6a_8$ and $a_1a_3a_5a_7a_9$ are SGB words?

33. [*20*] (Scot Morris.) Complete the following list of 26 interesting SGB words:

`about`, `bacon`, `faced`, `under`, `chief`, ..., `pizza`.

▶ **34.** [*21*] For each SGB word that doesn't include the letter `y`, obtain a 5-bit binary number by changing the vowels $\{$`a`, `e`, `i`, `o`, `u`$\}$ to 1 and the other letters to 0. What are the most common words for each of the 32 binary outcomes?

▶ **35.** [*26*] Sixteen well-chosen elements of `WORDS`(1000) lead to the branching pattern



,

codewords, b-ary
distance of a code
geometric net
parity bits
latin square
word square
dictionaries
palindromes
mirror pairs
alphabetic order
lexicographically
McManus
arithmetic progression
perfect shuffle
Morris
vowels
consonants
complete binary trie

which is a complete binary trie of words that begin with the letter `s`. But there's no such pattern of words beginning with `a`, even if we consider the full collection `WORDS`(5757).

What letters of the alphabet can be used as the starting letter of sixteen words that form a complete binary trie within `WORDS`(n), given n?

36. [*M17*] Explain the symmetries that appear in the word cube (10). Also show that two more such cubes can be obtained by changing only the two words {`stove`, `event`}.

37. [*20*] Which vertices of the graph *words*(5757, 0, 0, 0) have maximum degree?

38. [*22*] Using the digraph rule in (14), change `tears` to `smile` in just three steps, *without computer assistance.*

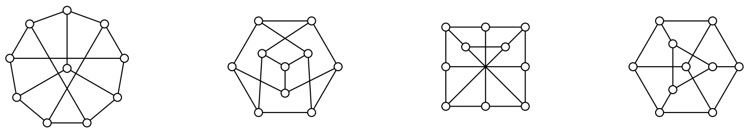
39. [*M00*] Is $G \setminus e$ an induced subgraph of G? Is it a spanning subgraph?

40. [*M15*] How many (a) spanning (b) induced subgraphs does a graph $G = (V, E)$ have, when $|V| = n$ and $|E| = e$?

41. [*M10*] For which integers n do we have (a) $K_n = P_n$? (b) $K_n = C_n$?

42. [*15*] (D. H. Lehmer.) Let G be a graph with 13 vertices, in which every vertex has degree 5. Make a nontrivial statement about G.

43. [*23*] Are any of the following graphs the same as the Petersen graph?



44. [*M23*] How many symmetries does Chvátal's graph have? (See Fig. 2(f).)

45. [*20*] Find an easy way to 4-color the planar graph (17). Would 3 colors suffice?

46. [*M25*] Let G be a graph with $n \geq 3$ vertices, defined by a planar diagram that is "maximal," in the sense that no additional lines can be drawn between nonadjacent vertices without crossing an existing edge.

a) Prove that the diagram partitions the plane into regions that each have exactly three vertices on their boundary. (One of these regions is the set of all points that lie outside the diagram.)

b) Therefore G has exactly $3n - 6$ edges.

47. [*M22*] Prove that the complete bigraph $K_{3,3}$ isn't planar.

48. [*M25*] Complete the proof of Theorem B by showing that the stated procedure never gives the same color to two adjacent vertices.

49. [*18*] Draw diagrams of all the cubic graphs with at most 6 vertices.

50. [*M24*] Find all bipartite graphs that can be 3-colored in exactly 24 ways.

▸ **51.** [*M22*] Given a geometric net as described in exercise 21, construct the bipartite graph whose vertices are the points p and the lines L of the net, with $p \mathbin{—} L$ if and only if $p \in L$. What is the *girth* of this graph?

52. [*M16*] Find a simple inequality that relates the diameter of a graph to its girth. (How small can the diameter be, if the girth is large?)

53. [*15*] Which of the words `world` and `happy` belongs to the giant component of the graph *words*(5757, 0, 0, 0)?

complete binary trie
trie
symmetries
word cube
degree
smile
graphs–
Lehmer
Petersen graph
symmetries
Chvátal's graph
contiguous USA+
planar
maximal planar graph
cubic graphs
geometric net
bipartite graph
girth
diameter
girth
words
giant component

▶ **54.** [*21*] The 49 postal codes in graph (17) are `AL`, `AR`, `AZ`, `CA`, `CO`, `CT`, `DC`, `DE`, `FL`, `GA`, `IA`, `ID`, `IL`, `IN`, `KS`, `KY`, `LA`, `MA`, `MD`, `ME`, `MI`, `MN`, `MO`, `MS`, `MT`, `NC`, `ND`, `NE`, `NH`, `NJ`, `NM`, `NV`, `NY`, `OH`, `OK`, `OR`, `PA`, `RI`, `SC`, `SD`, `TN`, `TX`, `UT`, `VA`, `VT`, `WA`, `WI`, `WV`, `WY`, in alphabetical order.

a) Suppose we consider two states to be adjacent if their postal codes agree in one place (namely `AL` — `AR` — `OR` — `OH`, etc.). What are the components of this graph?

b) Now form a directed graph with `XY` $\longrightarrow$ `YZ` (for example, `AL` $\longrightarrow$ `LA` $\longrightarrow$ `AR`, etc.). What are the *strongly connected components* of this digraph? (See Section 2.3.4.2.)

c) The United States has additional postal codes `AA`, `AE`, `AK`, `AP`, `AS`, `FM`, `GU`, `HI`, `MH`, `MP`, `PW`, `PR`, `VI`, besides those in (17). Reconsider question (b), using all 62 codes.

55. [*M20*] How many edges are in the complete k-partite graph $K_{n_1,\ldots,n_k}$?

▶ **56.** [*M10*] True or false: A multigraph is a graph if and only if the corresponding digraph is simple.

57. [*M10*] True or false: Vertices u and v are in the same connected component of a directed graph if and only if either $d(u,v) < \infty$ or $d(v,u) < \infty$.

58. [*M17*] Describe all (a) graphs (b) multigraphs that are regular of degree 2.

▶ **59.** [*M23*] A *tournament* of order n is a digraph on n vertices that has exactly $\binom{n}{2}$ arcs, either $u \longrightarrow v$ or $v \longrightarrow u$ for every pair of distinct vertices $\{u,v\}$.

a) Prove that every tournament contains an oriented spanning path $v_1 \longrightarrow \cdots \longrightarrow v_n$.

b) Consider the tournament on vertices $\{0,1,2,3,4\}$ for which $u \longrightarrow v$ if and only if $(u-v) \bmod 5 \geq 3$. How many oriented spanning paths does it have?

c) Is $K_n^{\rightarrow}$ the only tournament of order n that has a unique oriented spanning path?

▶ **60.** [*M22*] Let u be a vertex of greatest out-degree in a tournament, and let v be any other vertex. Prove that $d(u,v) \leq 2$.

61. [*M16*] Construct a digraph that has k walks of length k from vertex 1 to vertex 2.

62. [*M21*] A *permutation digraph* is a directed graph in which every vertex has out-degree 1 and in-degree 1; therefore its components are oriented cycles. If it has n vertices and k components, we call it *even* if $n-k$ is even, *odd* if $n-k$ is odd.

a) Let G be a directed graph with adjacency matrix A. Prove that the number of spanning permutation digraphs of G is per A, the permanent of A.

b) Interpret the determinant, det A, in terms of spanning permutation digraphs.

63. [*M23*] Let G be a graph of girth g in which every vertex has at least d neighbors. Prove that G has at least N vertices, where

$$N = \begin{cases} 1 + \sum_{0\leq k<t} d(d-1)^k, & \text{if } g = 2t+1; \\ 1 + (d-1)^t + \sum_{0\leq k<t} d(d-1)^k, & \text{if } g = 2t+2. \end{cases}$$

▶ **64.** [*M21*] Continuing exercise 63, show that there's a *unique* graph of girth 4, minimum degree d, and order $2d$, for each $d \geq 2$.

▶ **65.** [*HM31*] Suppose graph G has girth 5, minimum degree d, and $N = d^2+1$ vertices.

a) Prove that the adjacency matrix A of G satisfies the equation $A^2+A = (d-1)I+J$.

b) Since A is a symmetric matrix, it has N orthogonal eigenvectors x_j, with corresponding eigenvalues λ_j, such that $Ax_j = \lambda_j x_j$ for $1 \leq j \leq N$. Prove that each λ_j is either d or $(-1 \pm \sqrt{4d-3})/2$.

c) Show that if $\sqrt{4d-3}$ is irrational, then $d = 2$. *Hint:* $\lambda_1+\cdots+\lambda_N = \text{trace}(A) = 0$.

d) And if $\sqrt{4d-3}$ is rational, $d \in \{3, 7, 57\}$.

66. [*M30*] Continuing exercise 65, construct such a graph when $d = 7$.

postal codes
components
strongly connected components
complete k-partite graph
multigraph
digraph
simple
tournament
oriented spanning path
$K_n^{\rightarrow}$
transitive tournament
out-degree
permutation digraph
components
oriented cycles
even
odd
permanent
girth
adjacency matrix
eigenvectors
eigenvalues

67. [*M48*] Is there a regular graph of degree 57, order 3250, and girth 5?

68. [*M20*] How many different adjacency matrices does a graph G on n vertices have?

▶ **69.** [*20*] Extending (31), explain how to calculate both out-degree `ODEG(v)` and in-degree `IDEG(v)` for *all* vertices v in a graph that has been represented in SGB format.

▶ **70.** [*M20*] How often is each step of Algorithm B performed, when that algorithm successfully 2-colors a graph with m arcs and n vertices?

71. [*26*] Implement Algorithm B for the `MMIX` computer, using the `MMIXAL` assembly language. Assume that, when your program begins, register `v0` points to the first vertex node and register `n` contains the number of vertices.

▶ **72.** [*M22*] When `COLOR(v)` is set in step B6, call u the *parent* of v; but when `COLOR(w)` is set in step B3, say that w has no parent. Define the *ancestors* of vertex v, recursively, to be v together with the ancestors of v's parent (if any).

a) Prove that if v is below u in the stack during Algorithm B, the parent of v is an ancestor of u.

b) Furthermore, if `COLOR(v)` $=$ `COLOR(u)` in step B6, v is currently in the stack.

c) Use these facts to extend Algorithm B so that, if the given graph is not bipartite, the names of vertices in a cycle of odd length are output.

73. [*15*] What's another name for $random\_graph(10, 45, 0, 0, 0, 0, 0, 0, 0, 0)$?

74. [*21*] What vertex of $roget(1022, 0, 0, 0)$ has the largest out-degree?

75. [*22*] The SGB graph generator $board(n_1, n_2, n_3, n_4, p, w, o)$ creates a graph whose vertices are the t-dimensional integer vectors $(x_1, \ldots, x_t)$ for $0 \le x_i < b_i$, determined by the first four parameters (n_1, n_2, n_3, n_4) as follows: Set $n_5 \leftarrow 0$ and let $j \ge 0$ be minimum such that $n_{j+1} \le 0$. If $j = 0$, set $b_1 \leftarrow b_2 \leftarrow 8$ and $t \leftarrow 2$; this is the default 8×8 board. Otherwise if $n_{j+1} = 0$, set $b_i \leftarrow n_i$ for $1 \le i \le j$ and $t = j$. Finally, if $n_{j+1} < 0$, set $t \leftarrow |n_{j+1}|$, and set b_i to the ith element of the periodic sequence $(n_1, \ldots, n_j, n_1, \ldots, n_j, n_1, \ldots)$. (For example, the specification $(n_1, n_2, n_3, n_4) = (2, 3, 5, -7)$ is about as tricky as you can get; it produces a 7-dimensional board with $(b_1, \ldots, b_7) = (2, 3, 5, 2, 3, 5, 2)$, hence a graph with $2 \cdot 3 \cdot 5 \cdot 2 \cdot 3 \cdot 5 \cdot 2 = 1800$ vertices.)

The remaining parameters (p, w, o), for "piece, wrap, and orientation," determine the arcs of the graph. Suppose first that $w = o = 0$. If $p > 0$, we have $(x_1, \ldots, x_t) \longrightarrow (y_1, \ldots, y_t)$ if and only if $y_i = x_i + \delta_i$ for $1 \le i \le t$, where $(\delta_1, \ldots, \delta_t)$ is an integer solution to the equation $\delta_1^2 + \cdots + \delta_t^2 = |p|$. And if $p < 0$, we allow also $y_i = x_i + k\delta_i$ for $k \ge 1$, corresponding to k moves in the same direction.

If $w \ne 0$, let $w = (w_t \ldots w_1)_2$ in binary notation. Then we allow "wraparound," $y_i = (x_i + \delta_i) \bmod b_i$ or $y_i = (x_i + k\delta_i) \bmod b_i$, in each coordinate i for which $w_i = 1$.

If $o \ne 0$, the graph is directed; offsets $(\delta_1, \ldots, \delta_t)$ produce arcs only when they are lexicographically greater than $(0, \ldots, 0)$. But if $o = 0$, the graph is undirected.

Find settings of $(n_1, n_2, n_3, n_4, p, w, o)$ for which *board* will produce the following fundamental graphs: (a) the complete graph K_n; (b) the path P_n; (c) the cycle C_n; (d) the transitive tournament $K_n^{\rightarrow}$; (e) the oriented path $P_n^{\rightarrow}$; (f) the oriented cycle $C_n^{\rightarrow}$; (g) the $m \times n$ grid $P_m \,\square\, P_n$; (h) the $m \times n$ cylinder $P_m \,\square\, C_n$; (i) the $m \times n$ torus $C_m \,\square\, C_n$; (j) the $m \times n$ rook graph $K_m \,\square\, K_n$; (k) the $m \times n$ directed torus $C_m^{\rightarrow} \,\square\, C_n^{\rightarrow}$; (l) the null graph $\overline{K_n}$; (m) the n-cube $P_2 \,\square \cdots \square\, P_2$ with 2^n vertices.

76. [*20*] Can $board(n_1, n_2, n_3, n_4, p, w, o)$ produce loops, or parallel (repeated) edges?

77. [*M20*] If graph G has diameter ≥ 3, prove that $\overline{G}$ has diameter ≤ 3.

girth
adjacency matrices
out-degree
in-degree
SGB format
analysis of alg
depth-first search
MMIX
MMIXAL
bipartite
random_graph
roget
graph generator
board
chess, generalized pieces
wraparound
lexicographically
complete graph
path
cycle
transitive tournament
oriented path
oriented cycle
$m \times n$ grid
$m \times n$ cylinder
$m \times n$ torus
$m \times n$ rook graph
$m \times n$ directed torus
null graph
n-cube
loops
parallel (repeated) edges
multigraphs
diameter
complement

78. [*M26*] Let $G = (V, E)$ be a graph with $|V| = n$ and $G \cong \overline{G}$. (In other words, G is *self-complementary*: There's a permutation φ of V such that $u \mathrel{-\!\!-} v$ if and only if $\varphi(u) \not\mathrel{-\!\!-} \varphi(v)$ and $u \neq v$. We can imagine that the edges of K_n have been painted black or white; the white edges define a graph that's isomorphic to the graph of black edges.)

a) Prove that $n \bmod 4 = 0$ or 1. Draw diagrams for all such graphs with $n < 8$.
b) Prove that if $n \bmod 4 = 0$, every cycle of the permutation φ has a length that is a multiple of 4.
c) Conversely, every permutation φ with such cycles arises in some such graph G.
d) Extend these results to the case $n \bmod 4 = 1$.

▶ **79.** [*M22*] Given $k \geq 0$, construct a graph on the vertices $\{0, 1, \ldots, 4k\}$ that is both regular and self-complementary.

▶ **80.** [*M22*] A self-complementary graph must have diameter 2 or 3, by exercise 77. Given $k \geq 2$, construct self-complementary graphs of both possible diameters, when (a) $V = \{1, 2, \ldots, 4k\}$; (b) $V = \{0, 1, 2, \ldots, 4k\}$.

81. [*20*] The complement of a simple digraph without loops is defined by extending (35) and (36), so that we have $u \to v$ in $\overline{D}$ if and only if $u \neq v$ and $u \not\to v$ in D. What are the self-complementary digraphs of order 3?

82. [*M21*] Are the following statements about line graphs true or false?

a) If G is contained in G', then $L(G)$ is an induced subgraph of $L(G')$.
b) If G is a regular graph, so is $L(G)$.
c) $L(K_{m,n})$ is regular, for all $m, n > 0$.
d) $L(K_{m,n,r})$ is regular, for all $m, n, r > 0$.
e) $L(K_{m,n}) \cong K_m \,\square\, K_n$.
f) $L(K_4) \cong K_{2,2,2}$.
g) $L(P_{n+1}) \cong P_n$.
h) The graphs G and $L(G)$ both have the same number of components.

83. [*16*] Draw the graph $\overline{L(K_5)}$.

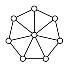
▶ **84.** [*M21*] Is $L(K_{3,3})$ self-complementary?

85. [*M22*] (O. Ore, 1962.) For which graphs G do we have $G \cong L(G)$?

86. [*M20*] (R. J. Wilson.) Find a graph G of order 6 for which $\overline{G} \cong L(G)$.

87. [*20*] Is the Petersen graph (a) 3-colorable? (b) 3-edge-colorable?

88. [*M20*] The graph $W_n = K_1 \mathrel{-\!\!-} C_{n-1}$ is called the *wheel* of order n, when $n \geq 4$. How many cycles does it contain as subgraphs?



W_8

89. [*M20*] Prove the associative laws, (42) and (43).

▶ **90.** [*M24*] A graph is called a *cograph* if it can be constructed algebraically from 1-element graphs by means of complementation and/or direct sum operations. For example, there are four nonisomorphic graphs of order 3, and they all are cographs: $\overline{K_3} = K_1 \oplus K_1 \oplus K_1$ and its complement, K_3; $\overline{K_{1,2}} = K_1 \oplus K_2$ and its complement, $K_{1,2}$, where $K_2 = \overline{K_1 \oplus K_1}$.

Exhaustive enumeration shows that there are 11 nonisomorphic graphs of order 4. Give algebraic formulas to prove that 10 of them are cographs. Which one isn't?

▶ **91.** [*20*] Draw diagrams for the 4-vertex graphs (a) $K_2 \,\square\, K_2$; (b) $K_2 \otimes K_2$; (c) $K_2 \boxtimes K_2$; (d) $K_2 \vartriangle K_2$; (e) $K_2 \circ K_2$; (f) $\overline{K_2} \circ K_2$; (g) $K_2 \circ \overline{K_2}$.

92. [*21*] The five types of graph products defined in the text work fine for simple digraphs as well as for ordinary graphs. Draw diagrams for the 4-vertex digraphs (a) $\vec{K_2} \,\square\, \vec{K_2}$; (b) $\vec{K_2} \otimes \vec{K_2}$; (c) $\vec{K_2} \boxtimes \vec{K_2}$; (d) $\vec{K_2} \vartriangle \vec{K_2}$; (e) $\vec{K_2} \circ \vec{K_2}$.

self-complementary
self-complementary graph
diameter
complement of a simple digraph
line graphs
induced subgraph
regular graph
components
Ore
Wilson
Petersen graph
edge-colorable
wheel
cycles
associative laws
cograph
complementation
direct sum
graphs of order 4
graph algebra++
graph products+
Cartesian product++
direct product+
strong product++
odd product+
lexicographic product+
digraphs

93. [*15*] Which of the five graph products takes K_m and K_n into K_{mn}?

94. [*10*] Are the SGB *words* graphs induced subgraphs of $P_{26} \,\square\, P_{26} \,\square\, P_{26} \,\square\, P_{26} \,\square\, P_{26}$?

95. [*M20*] If vertex u of G has degree d_u and vertex v of H has degree d_v, what is the degree of vertex (u,v) in (a) $G \square H$? (b) $G \otimes H$? (c) $G \boxtimes H$? (d) $G \triangle H$? (e) $G \circ H$?

▶ **96.** [*M22*] Let A be an $m \times m'$ matrix with $a_{uu'}$ in row u and column u'; let B be an $n \times n'$ matrix with $b_{vv'}$ in row v and column v'. The *direct product* $A \otimes B$ is an $mn \times m'n'$ matrix with $a_{uu'}b_{vv'}$ in row (u,v) and column (u',v'). Thus $A \otimes B$ is the adjacency matrix of $G \otimes H$, if A and B are the adjacency matrices of G and H.

Find analogous formulas for the adjacency matrices of (a) $G \,\square\, H$; (b) $G \boxtimes H$; (c) $G \triangle H$; (d) $G \circ H$.

97. [*M25*] Find as many interesting algebraic relations between graph sums and products as you can. (For example, the distributive law $(A \oplus B) \otimes C = (A \otimes C) \oplus (B \otimes C)$ for direct sums and products of matrices implies that $(G \oplus G') \otimes H = (G \otimes H) \oplus (G' \otimes H)$. We also have $\overline{K_m} \,\square\, H = H \oplus \cdots \oplus H$, with m copies of H, etc.)

98. [*M20*] If the graph G has k components and the graph H has l components, how many components are in the graphs $G \,\square\, H$ and $G \boxtimes H$?

99. [*M20*] Let $d_G(u,u')$ be the distance from vertex u to vertex u' in graph G. Prove that $d_{G \square H}((u,v),(u',v')) = d_G(u,u') + d_H(v,v')$, and find a similar formula for $d_{G \boxtimes H}((u,v),(u',v'))$.

100. [*M21*] For which connected graphs is $G \otimes H$ connected?

▶ **101.** [*M25*] Find all connected graphs G and H such that $G \,\square\, H \cong G \otimes H$.

102. [*M20*] What's a simple algebraic formula for the graph of *king moves* (which take one step horizontally, vertically, or diagonally) on an $m \times n$ board?

103. [*20*] Complete tableau (54). Also apply Algorithm H to the sequence 866444444.

104. [*18*] Explain the manipulation of variables i, t, and r in steps H3 and H4.

105. [*M34*] Suppose $d_1 \geq \cdots \geq d_n \geq 0$, and let $c_1 \geq \cdots \geq c_{d_1}$ be its conjugate as in Algorithm H. Prove that $d_1 \ldots d_n$ is graphical if and only if $d_1 + \cdots + d_n$ is even and $d_1 + \cdots + d_k \leq c_1 + \cdots + c_k - k$ for $1 \leq k \leq s$, where s is maximal such that $d_s \geq s$.

106. [*20*] True or false: If $d_1 = \cdots = d_n = d < n$ and nd is even, Algorithm H constructs a *connected* graph.

107. [*M21*] Prove that the degree sequence $d_1 \ldots d_n$ of a self-complementary graph satisfies $d_j + d_{n+1-j} = n - 1$ and $d_{2j-1} = d_{2j}$ for $1 \leq j \leq n/2$.

▶ **108.** [*M23*] Design an algorithm analogous to Algorithm H that constructs a *simple directed graph* on vertices $\{1, \ldots, n\}$, having specified values d_k^- and d_k^+ for the in-degree and out-degree of each vertex k, whenever at least one such graph exists.

109. [*M20*] Design an algorithm analogous to Algorithm H that constructs a *bipartite graph* on vertices $\{1, \ldots, m+n\}$, having specified degrees d_k for each vertex k when possible; all edges $j \,—\, k$ should have $j \leq m$ and $k > m$.

110. [*M22*] Without using Algorithm H, show by a direct construction that the sequence $d_1 \ldots d_n$ is graphical when $n > d_1 \geq \cdots \geq d_n \geq d_1 - 1$ and $d_1 + \cdots + d_n$ is even.

▶ **111.** [*25*] Let G be a graph on vertices $V = \{1, \ldots, n\}$, with d_k the degree of k and $\max(d_1, \ldots, d_n) = d$. Prove that there's an integer N with $n \leq N \leq 2n$ and a graph H on vertices $\{1, \ldots, N\}$, such that H is regular of degree d and $H \,|\, V = G$. Explain how to construct such a regular graph with N as small as possible.

words
induced subgraphs
five-letter words
degree
direct product
adjacency matrix
distributive law
direct sums
distance
king moves
conjugate
graphical
connected
degree sequence
self-complementary graph
simple directed graph
in-degree
out-degree
bipartite graph

▶ **112.** [*20*] Does the network *miles*$(128,0,0,0,0,127,0)$ have three equidistant cities? If not, what three cities come closest to an equilateral triangle?

113. [*05*] When H is a hypergraph with m edges and n vertices, how many rows and columns does its incidence matrix have?

114. [*M20*] Suppose the multigraph (26) is regarded as a hypergraph. What is the corresponding incidence matrix? What is the corresponding bipartite graph?

▶ **115.** [*M20*] When B is the incidence matrix of a graph G, explain the significance of the symmetric matrices B^TB and BB^T.

116. [*M17*] Describe the edges of the complete bipartite r-uniform hypergraph $K_{m,n}^{(r)}$.

117. [*M22*] How many nonisomorphic 1-uniform hypergraphs have m edges and n vertices? (Edges may be repeated.) List them all when $m=4$ and $n=3$.

118. [*M20*] A "hyperforest" is a hypergraph that contains no cycles. If a hyperforest has m edges, n vertices, and p components, what's the sum of the degrees of its vertices?

119. [*M18*] What hypergraph corresponds to (60) without the final term $(\bar{x}_1 \vee \bar{x}_2 \vee \bar{x}_3)$?

120. [*M20*] Define *directed hypergraphs*, by generalizing the concept of directed graphs.

121. [*M19*] Given a hypergraph $H=(V,E)$, let $I(H)=(V,F)$, where F is the family of all maximal independent sets of H. Express $\chi(H)$ in terms of $|V|$, $|F|$, and $\alpha(I(H)^T)$.

▶ **122.** [*M24*] Find a maximum independent set and a minimum coloring of the following triple systems: (a) the hypergraph (56); (b) the dual of the Petersen graph.

123. [*17*] Show that the optimum colorings of $K_n \,\square\, K_n$ are equivalent to the solutions of a famous combinatorial problem.

124. [*M22*] What is the chromatic number of the Chvátal graph, Fig. 2(f)?

125. [*M48*] For what values of g is there a 4-regular, 4-chromatic graph of girth g?

▶ **126.** [*M22*] Find optimum colorings of the "kingwise torus," $C_m \boxtimes C_n$, when $m,n \ge 3$.

127. [*M22*] Prove that (a) $\chi(G)+\chi(\overline{G}) \le n+1$ and (b) $\chi(G)\chi(\overline{G}) \ge n$ when G is a graph of order n, and find graphs for which equality holds.

128. [*M18*] Express $\chi(G \,\square\, H)$ in terms of $\chi(G)$ and $\chi(H)$, when G and H are graphs.

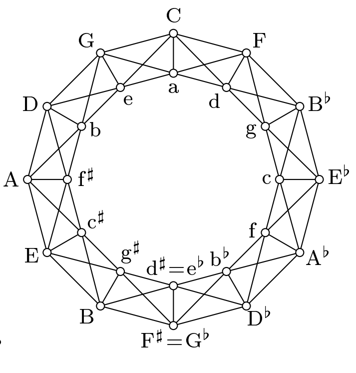
129. [*23*] Describe the maximal cliques of the 8×8 queen graph (37).

130. [*M20*] How many maximal cliques are in a complete k-partite graph?

131. [*M30*] Let $N(n)$ be the largest number of maximal cliques that an n-vertex graph can have. Prove that $3^{\lfloor n/3 \rfloor} \le N(n) \le 3^{\lceil n/3 \rceil}$.

▶ **132.** [*M20*] We call G *tightly colorable* if $\chi(G)=\omega(G)$. Prove that $\chi(G \boxtimes H) = \chi(G)\chi(H)$ whenever G and H are tightly colorable.

133. [*21*] The "musical graph" illustrated here provides a nice way to review numerous definitions that were given in this section, because its properties are easily analyzed. Determine its (a) order; (b) size; (c) girth; (d) diameter; (e) independence number, $\alpha(G)$; (f) chromatic number, $\chi(G)$; (g) edge-chromatic number, $\chi(L(G))$; (h) clique number, $\omega(G)$; (i) algebraic formula as a product of well-known smaller graphs. What is the size of (j) a minimum vertex cover? (k) a maximum matching? Is G (l) regular? (m) planar? (n) connected? (o) directed? (p) a free tree? (q) Hamiltonian?

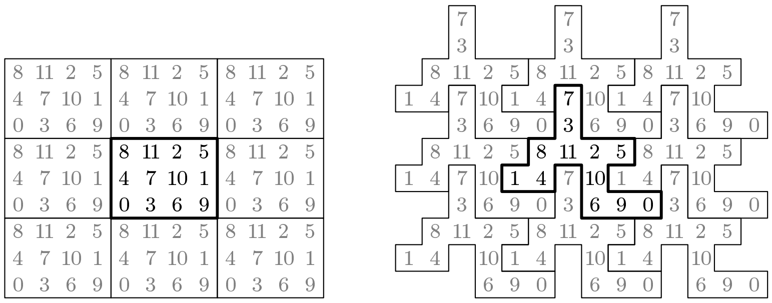


network
miles
equidistant cities
hypergraph
incidence matrix
multigraph
bipartite graph
incidence matrix
complete bipartite r-uniform hypergraph
$K_{m,n}^{(r)}$
hyperforest
cycles
degrees
directed hypergraphs
maximal independent sets
triple systems
dual
Petersen graph
Chvátal graph
kingwise torus
queen graph
complete k-partite graph
tightly colorable
musical graph
order
size
girth
diameter
independence number
chromatic number
edge-chromatic number
clique number
minimum vertex cover
maximum matching
regular
planar
connected
directed
free tree
Hamiltonian

134. [*M22*] How many automorphisms does the musical graph have?

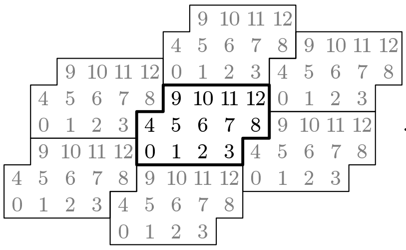
▸ **135.** [*HM26*] Suppose a composer takes a random walk in the musical graph, starting at vertex C and then making five equally likely choices at each step. Show that after an even number of steps, the walk is more likely to end at vertex C than at any other vertex. What is the exact probability of going from C to C in a 12-step walk?

136. [*HM23*] A *Cayley digraph* is a directed graph whose vertices V are the elements of a group and whose arcs are $v \longrightarrow v\alpha_j$ for $1 \le j \le d$ and all vertices v, where $(\alpha_1, \ldots, \alpha_d)$ are fixed elements of the group. A *Cayley graph* is a Cayley digraph that is also a graph. Is the Petersen graph a Cayley graph?



▸ **137.** [*M25*] (*Generalized toruses.*) An $m \times n$ torus can be regarded as a tiling of the plane. For example, we can imagine that infinitely many copies of the 3×4 torus in (50) have been placed together gridwise, as indicated in the left-hand illustration above; from each vertex we can move north, south, east, or west to another vertex of the torus. The vertices have been numbered here so that a northward move from v goes to $(v{+}4) \bmod 12$, and an eastward move to $(v{+}3) \bmod 12$, etc. The right-hand illustration shows the same torus, but with a differently shaped tile; *any* way to choose twelve cells numbered $\{0, 1, \ldots, 11\}$ will tile the plane, with exactly the same underlying graph.

Shifted copies of a single shape will also tile the plane if they form a *generalized* torus, in which cell (x, y) corresponds to the same vertex as cells $(x + a, y + b)$ and $(x + c, y + d)$, where (a, b) and (c, d) are integer vectors and $n = ad - bc > 0$. The generalized torus will then have n points. These vectors (a, b) and (c, d) are $(4, 0)$ and $(0, 3)$ in the 3×4 example above; and when they are respectively $(5, 2)$ and $(1, 3)$ we get

 .

Here $n = 13$, and a northward move from v goes to $(v + 4) \bmod 13$; an eastward move goes to $(v + 1) \bmod 13$.

Prove that if $\gcd(a, b, c, d) = 1$, the vertices of such a generalized torus can always be assigned integer labels $\{0, 1, \ldots, n{-}1\}$ in such a way that the neighbors of v are $(v \pm p) \bmod n$ and $(v \pm q) \bmod n$, for some integers p and q.

automorphisms
random walk
musical graph
walk
Cayley digraph
Petersen graph
–graphs
generalized toruses+
translation, tiling by+
tiling of the plane

138. [*HM27*] Continuing exercise 137, what is a good way to label k-dimensional vertices $x = (x_1, \ldots, x_k)$, when integer vectors α_j are given such that each vector x is equivalent to $x + \alpha_j$ for $1 \le j \le k$? Illustrate your method in the case $k = 3$, $\alpha_1 = (3, 1, 1)$, $\alpha_2 = (1, 3, 1)$, $\alpha_3 = (1, 1, 3)$.

▸ **139.** [*M22*] Let H be a fixed graph of order h, and let $\#(H{:}G)$ be the number of times that H occurs as an induced subgraph of a given graph G. If G is chosen at random from the set of all $2^{n(n-1)/2}$ graphs on the vertices $V = \{1, 2, \ldots, n\}$, what is the average value of $\#(H{:}G)$ when H is (a) K_h; (b) P_h, for $h > 1$; (c) C_h, for $h > 2$; (d) arbitrary?

140. [*M30*] A graph G is called *proportional* if its induced subgraph counts $\#(K_3{:}G)$, $\#(\overline{K_3}{:}G)$, and $\#(P_3{:}G)$ each agree with the expected values derived in exercise 139.

a) Show that the wheel graph W_8 of exercise 88 is proportional in this sense.

b) Prove that G is proportional if and only if $\#(K_3{:}G) = \frac{1}{8}\binom{n}{3}$ and the degree sequence $d_1 \ldots d_n$ of its vertices satisfies the identities

$$d_1 + \cdots + d_n = \binom{n}{2}, \qquad d_1^2 + \cdots + d_n^2 = \frac{n}{2}\binom{n}{2}. \qquad (*)$$

141. [*26*] The conditions of exercise 140(b) can hold only if $n \bmod 16 \in \{0, 1, 8\}$. Write a program to find all of the proportional graphs that have $n = 8$ vertices.

142. [*M30*] (S. Janson and J. Kratochvíl, 1991.) Prove that no graph G on 4 or more vertices can be "extraproportional," in the sense that its subgraph counts $\#(H{:}G)$ agree with the expected values in exercise 139 for each of the eleven nonisomorphic graphs H of order 4. *Hint:* $(n-3)\#(K_3{:}G) = 4\#(K_4{:}G) + 2\#(K_{1,1,2}{:}G) + \#(K_1{\oplus}K_3{:}G)$.

▸ **143.** [*M25*] Let A be any matrix with $m > 1$ distinct rows, and $n \ge m$ columns. Prove that at least one column of A can be deleted, without making any two rows equal.

▸ **144.** [*21*] Let X be an $m \times n$ matrix whose entries x_{ij} are either 0, 1, or $*$. A "completion" of X is a matrix X^* in which every $*$ has been replaced by either 0 or 1. Show that the problem of finding a completion with fewest distinct rows is equivalent to the problem of finding the chromatic number of a graph.

▸ **145.** [*25*] (R. S. Boyer and J. S. Moore, 1980.) Suppose the array $a_1 \ldots a_n$ contains a *majority element*, namely a value that occurs more than $n/2$ times. Design an algorithm that finds it after making fewer than n comparisons. *Hint:* If $n \ge 3$ and $a_{n-1} \ne a_n$, the majority element of $a_1 \ldots a_n$ is also the majority element of $a_1 \ldots a_{n-2}$.

induced subgraph
random
proportional
wheel graph
degree sequence
Janson
Kratochvíl
order 4
distinct rows
completion
chromatic number
coloring
don't cares
Boyer
Moore
majority element

Davies
Hall
Leahy
Skolem
Bang
Simpson
phi
spectrum

ANSWERS TO EXERCISES

Answer not a fool according to his folly,
lest thou also be like unto him.
— Proverbs 26:4

SECTION 7

1. Following the hint, we'll want the second '$4m-4$' to be immediately followed by the first '$2m-1$'. The desired arrangements can be deduced from the first four examples, given in hexadecimal notation: `231213`, `46171435623725`, `86a31b1368597a425b2479`, `ca8e531f1358ac7db9e6427f2469bd`. [R. O. Davies, *Math. Gazette* **43** (1959), 253–255.]

2. Such arrangements exist if and only if $n \bmod 4 = 0$ or 1. This condition is necessary because there must be an even number of odd items. And it is sufficient because we can place '00' in front of the solutions in the previous exercise.

Notes: This question was first raised by Marshall Hall in 1951, and solved the following year by F. T. Leahy, Jr., in unpublished work [Armed Forces Security Agency report 343 (28 January 1952)]. It was independently posed and resolved by T. Skolem and T. Bang, *Math. Scandinavica* **5** (1957), 57–58. For other intervals of numbers, see the complete solution by J. E. Simpson, *Discrete Math.* **44** (1983), 97–104.

3. Yes. For example, the cycle (0072362435714165) can't be broken up.

4. The kth occurrence of b is in position $\lfloor k\phi \rfloor$ from the left, and the kth occurrence of a is in position $\lfloor k\phi^2 \rfloor$. Clearly $\lfloor k\phi^2 \rfloor - \lfloor k\phi \rfloor = k$, because $\phi^2 = \phi + 1$. (The integers $\lfloor k\phi \rfloor$ form the "spectrum" of ϕ; see exercise 3.13 of *CMath*.)

5. $2n-k-1$ of the $\binom{2n}{2}$ equally likely pairs of positions satisfy the stated condition. If these probabilities were independent (but they aren't), the value of $2L_n$ would be

$$\binom{2n}{2,2,\ldots,2}\prod_{k=1}^{n}\Bigl((2n-1-k)/\tbinom{2n}{2}\Bigr) = \frac{(2n)!^2\,n(n-1)}{n!\,(2n)^{n+1}(2n-1)^{n+1}}$$
$$= \exp\Bigl(n\ln\frac{4n}{e^3} + \ln\sqrt{\frac{\pi e n}{2}} + O(n^{-1})\Bigr).$$

6. (a) When the products are expanded, we obtain a polynomial of $(2n-2)!/(n-2)!$ terms, each of degree $4n$. There's a term $x_1^2 \ldots x_{2n}^2$ for each Langford pairing; every other term has at least one variable of degree 1. Summing over $x_1, \ldots, x_{2n} \in \{-1,+1\}$ therefore cancels out all the bad terms, but gives 2^{2n} for the good terms. An extra factor of 2 arises because there are $2L_n$ Langford pairings (including left-right reversals).

(b) Let $f_k = \sum_{j=1}^{2n-k-1} x_j x_{j+k+1}$ be the main part of the kth factor. We can run through all 4^n cases $x_1, \ldots, x_{2n} \in \{-1,+1\}$ in Gray-code order (Algorithm 7.2.1.1L), negating only one of the x_j each time. A change in x_j causes at most two adjustments to each f_k; so each Gray-code step costs $O(n)$.

We needn't compute the sum exactly; it suffices to work mod 2^N, where 2^N comfortably exceeds $2^{2n+1}L_n$. Even better, when $n = 24$, would be to do the computations mod $2^{60}-1$, or mod both $2^{30}-1$ and $2^{30}+1$. One can also save $\lceil n/2 \rceil$ bits of precision by exploiting the fact that $f_k \equiv k+1$ (modulo 2).

(c) The third equality is actually valid only when $n \bmod 4 = 0$ or 3; but those are the interesting n's. The sum can be carried out in n phases, where phase p for $p < n$ involves the cases where $x_{n-1} = x_{n+2}$, $x_{n-2} = x_{n+3}$, $\ldots$, $x_{n-p+1} = x_{n+p}$, $x_{n-p} = x_n = x_{n+1} = +1$, and $x_{n+p+1} = -1$; it has an outer loop that chooses $(x_{n-p+1}, \ldots, x_{n-1})$ in all 2^{p-1} ways, and an inner loop that chooses $(x_1, \ldots, x_{n-p-1}, x_{n+p+2}, \ldots, x_{2n})$ in all $2^{2n-2p-2}$ ways. (The inner loop uses Gray binary code, preferably with "organ-pipe order" to prioritize the subscripts so that x_1 and x_{2n} vary most rapidly. The outer loop need not be especially efficient.) Phase n covers the 2^{n-1} palindromic cases with $x_j = x_{2n+1-j}$ for $1 \le j < n$ and $x_n = x_{n+1} = +1$. If s_p denotes the sum in phase p, then $s_1 + \cdots + s_{n-1} + \frac{1}{2}s_n = 2^{2n-2}L_n$.

A substantial fraction of the terms turn out be zero. For example, when $n = 16$, zeros appear about 76% of the time (in 408,838,754 cases out of $2^{29}+2^{14}$). This fact can be used to avoid many multiplications in the inner loop. (Only f_1, f_3, $\ldots$ can be zero.)

7. Let d_k be the number of incomplete pairs after k characters have been read; thus $d_0 = d_{2n} = 0$, and $d_k = d_{k-1} \pm 1$ for $1 \le k \le 2n$. The largest such sequence in which d_k never exceeds 6 is $(0,1,2,3,4,5,6,5,6,\ldots,5,6,5,4,3,2,1,0)$. This sequence has $\sum_{k=1}^{2n} d_k = 11n - 30$. But $\sum_{k=1}^{2n} d_k = \sum_{k=1}^{n}(k+1) = \binom{n+1}{2} + n$ in any Langford pairing. Hence $\binom{n+1}{2} + n \le 11n - 30$, and $n \le 15$. (In fact, width 6 is also impossible when $n = 15$. The largest and smallest possible width are unknown in general.)

8. There are no solutions when $n = 4$ or $n = 7$. When $n = 8$ there are four:

1317538642572468; 1418634753268257; 4275248635713168; 5286235743681417.

(This problem makes a pleasant mechanical puzzle, using gadgets of width $k+1$ and height $\lceil k/2 \rceil$ for piece k. In his original note [*Math. Gazette* **42** (1958), 228], C. Dudley Langford illustrated similar pieces, and exhibited a planar solution for $n = 12$. The question can be cast as an exact cover problem, with nonprimary columns representing places where two gadgets are not allowed to intersect; see exercise 7.2.2.1–00. Jean Brette has devised a somewhat similar puzzle, based on Skolem's variant of the problem and using width instead of planarity; he gave a copy to David Singmaster in 1992.)

9. Just three ways: 181915267285296475384639743, 191218246279458634753968357, 191618257269258476354938743 (and their reversals). [First found in 1969 by G. Baron; see *Combinatorial Theory and Its Applications* (Budapest: 1970), 81–92. The "dancing links" method of Section 7.2.2.1 resolves this question by traversing a search tree that has only 360 nodes, given an exact cover problem with 132 rows.]

10. For example, let `A` $= 12$, `K` $= 8$, `Q` $= 4$, `J` $= 0$, $\spadesuit = 4$, $\heartsuit = 3$, $\diamondsuit = 2$, $\clubsuit = 1$; add.

[In this connection, orthogonal latin squares equivalent to Fig. 1 were implicitly present already in medieval Islamic talismans illustrated by Ibn al-Hajj in his *Kitab Shumus al-Anwar* (Cairo: 1322); he also gave a 5×5 example. See E. Doutté, *Magie*

Gray-code
modular arithmetic
organ-pipe order
palindromic
puzzle
RainBones puzzle
Langford
exact cover problem
Brette
Skolem
width
Singmaster
Baron
dancing links
exact cover problem
Ibn al-Hajj
Islamic maths
Arabic maths
Doutté

et Religion dans l'Afrique du Nord (Algiers: 1909), 193–194, 214, 247; W. Ahrens, *Der Islam* **7** (1917), 228–238. See also an article on the history of latin squares being prepared by Lars D. Andersen.]

11. $\begin{pmatrix} d\gamma\text{א} & a\delta\text{ב} & b\beta\text{ג} & c\alpha\text{ד} \\ c\beta\text{ב} & b\alpha\text{א} & a\gamma\text{ד} & d\delta\text{ג} \\ a\alpha\text{ג} & d\beta\text{ד} & c\delta\text{א} & b\gamma\text{ב} \\ b\delta\text{ד} & c\gamma\text{ג} & d\alpha\text{ב} & a\beta\text{א} \end{pmatrix}$. [Joseph Sauveur presented the earliest known example of such squares in *Mémoires de l'Académie Royale des Sciences* (Paris, 1710), 92–138, §83.]

12. If n is odd, we can let $M_{ij} = (i-j) \bmod n$. But if n is even, there are no transversals: For if $\{(t_0{+}0) \bmod n, \ldots, (t_{n-1}{+}n{-}1) \bmod n\}$ is a transversal, we have $\sum_{k=0}^{n-1} t_k \equiv \sum_{k=0}^{n-1}(t_k + k)$ (modulo n), hence $\sum_{k=0}^{n-1} k = \frac{1}{2}n(n-1)$ is a multiple of n.

13. Replace each element l by $\lfloor l/5 \rfloor$ to get a matrix of 0s and 1s. Let the four quarters be named $\left(\begin{smallmatrix} A & B \\ C & D \end{smallmatrix}\right)$; then A and D each contain exactly k 1s, while B and C each contain exactly k 0s. Suppose the original matrix has ten disjoint transversals. If $k \le 2$, at most four of them go through a 1 in A or D, and at most four go through a 0 in B or C. Thus at least two of them hit only 0s in A and D, only 1s in B and C. But such a transversal has an even number of 0s (not five), because it intersects A and D equally often.

Similarly, a latin square of order $4m+2$ with an orthogonal mate must have more than m intruders in each of its $(2m+1) \times (2m+1)$ submatrices, under all renamings of the elements. [H. B. Mann, *Bull. Amer. Math. Soc.* (2) **50** (1944), 249–257.]

14. Cases (b) and (d) have no mates. Cases (a), (c), and (e) have respectively 2, 6, and 12265168(!), of which the lexicographically first and last are

(a)	(a)	(c)	(c)	(e)	(e)
0456987213	0691534782	0362498571	0986271435	0214365897	0987645321
1305629847	1308257964	1408327695	1354068792	1025973468	1795402638
2043798165	2169340578	2673519408	2741853960	2690587143	2506913874
3289176504	3250879416	3521970846	3572690814	3857694201	3154067289
4518263790	4587902631	4890253167	4630789251	4168730925	4231850967
5167432089,	5412763890;	5736841920,	5218947306;	5473829016,	5348276190.
6894015372	6945081327	6259784013	6095324178	6942158730	6820394715
7920341658	7836425109	7915602384	7869512043	7309216584	7069128543
8731504926	8723196045	8147036259	8407136529	8531402679	8412739056
9672850431	9074618253	9084165732	9123405687	9786041352	9673581402

Notes: Squares (a), (b), (c), and (d) were obtained from the decimal digits of π, e, γ, and ϕ, by discarding each digit that is inconsistent with a completed latin square. Although they aren't truly random, they're probably typical of 10×10 latin squares in general, roughly half of which appear to have orthogonal mates. Parker constructed square (e) in order to obtain an unusually large number of transversals; it has 5504 of them. (Euler had studied a similar example of order 6, therefore "just missing" the discovery of a 10×10 pair.)

15. Parker was dismayed to discover that none of the mates of square 14(e) are orthogonal to each other. With J. W. Brown and A. S. Hedayat [*J. Combinatorics, Inf. and System Sci.* **18** (1993), 113–115], he later found two $10{\times}10$s that have four disjoint common transversals (but not ten). [See also B. Ganter, R. Mathon, and A. Rosa, *Congressus Numerantium* **20** (1978), 383–398; **22** (1979), 181–204.] While pursuing an idea of L. Weisner [*Canadian Math. Bull.* **6** (1963), 61–63], the author accidentally noticed some squares that come even closer to a mutually orthogonal trio: The square below is orthogonal to its transpose; and it has five diagonally symmetric transversals, in cells $(0, p_0)$, …, $(9, p_9)$ for $p_0 \ldots p_9$ = 0132674598, 2301457689, 3210896745, 4897065312, and 6528410937, which are *almost* disjoint: They cover 49 cells.

$$L \;=\; \begin{pmatrix} 0234567891\\ 3192708546\\ 6528139407\\ 8753241960\\ 1689473025\\ 4970852613\\ 5047986132\\ 9416320758\\ 7361095284\\ 2805614379 \end{pmatrix} \;\perp\; \begin{pmatrix} 0368145972\\ 2157690438\\ 3925874160\\ 4283907615\\ 5712489306\\ 6034758291\\ 7891326054\\ 8549061723\\ 9406213587\\ 1670532849 \end{pmatrix} \;=\; L^T.$$

Extensive computations by B. D. McKay, A. Meynert, and W. Myrvold [*J. Comb. Designs* **15** (2007), 98–119] prove that no 10×10 latin square with nontrivial symmetry has two mates orthogonal to each other. Three mutually orthogonal latin squares are known to exist for all orders $n > 10$ [see S. M. P. Wang and R. M. Wilson, *Congressus Numerantium* **21** (1978), 688; D. T. Todorov, *Ars Combinatoria* **20** (1985), 45–47].

McKay
Meynert
Myrvold
Wang
Wilson
Todorov
Brualdi
Ryser
latin square
Græco-Latin squares
Hebraic
Rao

16. See R. A. Brualdi and H. J. Ryser, *Combinatorial Matrix Theory* (Cambridge University Press, 1991), §8.2.

17. (a) Let there be $3n$ columns r_j, c_j, v_j for $0 \le j < n$, and n^2 rows; row (i,j) has 1 in columns r_i, c_j, and v_l, where $l = L_{ij}$, for $0 \le i,j < n$.

(b) Let there be $4n^2$ columns r_{ij}, c_{ij}, x_{ij}, y_{ij} for $0 \le i,j < n$, and $n^3 - n^2 + n$ rows; row (i,j,k) has 1 in columns r_{ik}, c_{jk}, x_{ij}, and y_{lk}, where $l = L_{ij}$, for $0 \le i,j,k < n$ and $(i = k$ or $j > 0)$.

18. Given an orthogonal array A with rows A_i for $1 \le i \le m$, define latin square $L_i = (L_{ijk})$ for $1 \le i \le m-2$ by setting $L_{ijk} = A_{iq}$ when $A_{(m-1)q} = j$ and $A_{mq} = k$, for $0 \le j,k < n$. (The value of q is uniquely determined by the values of j and k.) Permuting the columns of the array does not change the corresponding latin squares.

This construction can also be reversed, to produce orthogonal arrays of order n from mutually orthogonal latin squares of order n. In exercise 11, for example, we can let $a = \alpha = \aleph = 0$, $b = \beta = \beth = 1$, $c = \gamma = \gimel = 2$, and $d = \delta = \daleth = 3$, obtaining

$$A = \begin{pmatrix} 3012210303211230\\ 2310102301323201\\ 0123103223013210\\ 0000111122223333\\ 0123012301230123 \end{pmatrix}.$$

(The concept of an orthogonal array is mathematically "cleaner" than the concept of orthogonal latin squares, because it accounts better for the underlying symmetries. Notice, for example, that an $n\times n$ matrix L is a latin square if and only if it is orthogonal to two particular non-latin squares, namely

$$L \perp \begin{pmatrix} 1 & 1 & \dots & 1\\ 2 & 2 & \dots & 2\\ \vdots & \vdots & \ddots & \vdots\\ n & n & \dots & n \end{pmatrix} \qquad\text{and}\qquad L \perp \begin{pmatrix} 1 & 2 & \dots & n\\ 1 & 2 & \dots & n\\ \vdots & \vdots & \ddots & \vdots\\ 1 & 2 & \dots & n \end{pmatrix}.$$

Therefore Latin squares, Græco-Latin squares, Hebraic-Græco-Latin squares, etc., are equivalent to orthogonal arrays of depth 3, 4, 5, ... Moreover, the orthogonal arrays considered here are merely the special case $t = 2$ and $\lambda = 1$ of a more general concept of n-ary $m \times \lambda n^t$ arrays having "strength t" and "index λ," introduced by C. R. Rao in *Proc. Edinburgh Math. Soc.* **8** (1949), 119–125; see the book *Orthogonal Arrays* by A. S. Hedayat, N. J. A. Sloane, and J. Stufken (Springer, 1999).)

19. We can rearrange the columns so that the first row is $0^n 1^n \ldots (n{-}1)^n$. Then we can renumber the elements of the other rows so that they begin with $01\ldots(n{-}1)$. The elements in each remaining column must then be distinct, in all rows but the first.

To achieve the upper bound when $n = p$, let each column be indexed by two numbers x and y, where $0 \le x, y < p$, and put the numbers y, x, $(x + y) \bmod p$, $(x+2y) \bmod p$, $\ldots$, $(x+(p{-}1)y) \bmod p$ into that column. For example, when $p = 5$ we get the following orthogonal array, equivalent to four mutually orthogonal latin squares:

$$\begin{pmatrix}
0\,0\,0\,0\,0\,1\,1\,1\,1\,1\,2\,2\,2\,2\,2\,3\,3\,3\,3\,3\,4\,4\,4\,4\,4\\
0\,1\,2\,3\,4\,0\,1\,2\,3\,4\,0\,1\,2\,3\,4\,0\,1\,2\,3\,4\,0\,1\,2\,3\,4\\
0\,1\,2\,3\,4\,1\,2\,3\,4\,0\,2\,3\,4\,0\,1\,3\,4\,0\,1\,2\,4\,0\,1\,2\,3\\
0\,1\,2\,3\,4\,2\,3\,4\,0\,1\,4\,0\,1\,2\,3\,1\,2\,3\,4\,0\,3\,4\,0\,1\,2\\
0\,1\,2\,3\,4\,3\,4\,0\,1\,2\,1\,2\,3\,4\,0\,4\,0\,1\,2\,3\,2\,3\,4\,0\,1\\
0\,1\,2\,3\,4\,4\,0\,1\,2\,3\,3\,4\,0\,1\,2\,2\,3\,4\,0\,1\,1\,2\,3\,4\,0
\end{pmatrix}.$$

[Essentially the same idea works when n is a prime power, using the finite field $\mathrm{GF}(p^e)$; see E. H. Moore, *American Journal of Mathematics* **18** (1896), 264–303, §15(l). These arrays are equivalent to finite *projective planes*; see Marshall Hall, Jr., *Combinatorial Theory* (Blaisdell, 1967), Chapters 12 and 13.]

20. Let $\omega = e^{2\pi i/n}$, and suppose $a_1 \ldots a_{n^2}$ and $b_1 \ldots b_{n^2}$ are the vectors in different rows. Then $a_1 b_1 + \cdots + a_{n^2} b_{n^2} = \sum_{0 \le j,k<n} \omega^{j+k} = 0$ because $\sum_{k=0}^{n-1} \omega^k = 0$.

21. (a) To show that equality-or-parallelism is an equivalence relation, we need to verify the transitive law: If $L \parallel M$ and $M \parallel N$ and $L \ne N$, then we must have $L \parallel N$. Otherwise there would be a point p with $L \cap N = \{p\}$, by (ii); and p would lie on two different lines parallel to M, contradicting (iii).

(b) Let $\{L_1, \ldots, L_n\}$ be a class of parallel lines, and assume that M is a line of another class. Then each L_j intersects M in a unique point p_j; and every point of M is encountered in this way, because every point of the geometry lies on exactly one line of each class, by (iii). Thus M contains exactly n points.

(c) We've already observed that every point belongs to m lines when there are m classes. If lines L, M, and N belong to three different classes, then M and N have the same number of points as the number of lines in L's class. So there's a common line size n, and in fact the total number of points is n^2. (Of course n might be infinite.)

22. Given an orthogonal array A of order n and depth m, define a geometric net with n^2 points and m classes of parallel lines by regarding the columns of A as points; line j of class k is the set of columns where symbol j appears in row k of A.

All finite geometric nets with $m \ge 3$ classes arise in this way. But a geometric net with only one class is trivially a partition of the points into disjoint subsets. A geometric net with $m = 2$ classes has nn' points (x, x'), where there are n lines '$x =$ constant' in one class and n' lines '$x' =$ constant' in the other. [For further information, see R. H. Bruck, *Canadian J. Math.* **3** (1951), 94–107; *Pacific J. Math.* **13** (1963), 421–457.]

23. (a) If $d(x, y) \le t$ and $d(x', y) \le t$ and $x \ne x'$, then $d(x, x') \le 2t$. Thus a code with distance $> 2t$ between codewords allows the correction of up to t errors — at least in principle, although the computations might be complex. Conversely, if $d(x, x') \le 2t$ and $x \ne x'$, there's an element y with $d(x, y) \le t$ and $d(x', y) \le t$; hence we can't reconstruct x uniquely when y is received.

(b, c) Let $m = r + 2$, and observe that a set of b^2 b-ary m-tuples has Hamming distance $\ge m - 1$ between all pairs of elements if and only if it forms the columns of a

Hedayat
Sloane
Stufken
finite field
Moore
projective planes
Hall
equivalence relation
transitive law
Bruck

b-ary orthogonal array of depth m. [See S. W. Golomb and E. C. Posner, *IEEE Trans.* **IT-10** (1964), 196–208. The literature of coding theory often denotes a code $C(b,n,r)$ of distance d by the symbol $(n+r, b^n, d)_b$. Thus, a b-ary orthogonal array of depth m is essentially an $(m, b^2, m-1)_b$ code.]

orthogonal array
Golomb
Posner
coding theory
majority
witness bits
Hsiao
Bossen
Chien
anagrams
Chaucer
Leighton
Gordon

24. (a) Suppose $x_j \neq x'_j$ for $1 \le j \le l$ and $x_j = x'_j$ for $l < j \le N$. We have $x = x'$ if $l = 0$. Otherwise consider the parity bits that correspond to the m lines through point 1. At most $l-1$ of those bits correspond to lines that touch the points $\{2, \ldots, l\}$. Hence x' has at least $m-(l-1)$ parity changes, and $d(x,x') \ge l + (m-(l-1)) = m+1$.

(b) Let l_{p1}, $\ldots$, l_{pm} be the index numbers of the lines through point p. After receiving a message $y_1 \ldots y_{N+R}$, compute x_p for $1 \le p \le N$ by taking the majority value of the $m+1$ "witness bits" $\{y_{p0}, \ldots, y_{pm}\}$, where $y_{p0} = y_p$ and

$$y_{pk} = \bigl(y_{N+l_{pk}} + \textstyle\sum\{y_j \mid j \neq p \text{ and point } j \text{ lies on line } l_{pk}\}\bigr) \bmod 2, \quad \text{for } 1 \le k \le m.$$

This method works because each received bit y_j affects at most one of the witness bits.

For example, in the 25-point geometry of exercise 19, suppose the parity bit $x_{26+5i+j}$ of each codeword corresponds to line j of row i, for $0 \le i \le 5$ and $0 \le j < 5$; thus $x_{26} = x_1 \oplus x_2 \oplus x_3 \oplus x_4 \oplus x_5$, $x_{27} = x_6 \oplus x_7 \oplus x_8 \oplus x_9 \oplus x_{10}$, $\ldots$, $x_{55} = x_5 \oplus x_6 \oplus x_{12} \oplus x_{18} \oplus x_{24}$. Given message $y_1 \ldots y_{55}$, we decode bit x_1 (say) by computing the majority of the seven bits y_1, $y_{26} \oplus y_2 \oplus y_3 \oplus y_4 \oplus y_5$, $y_{31} \oplus y_6 \oplus y_{11} \oplus y_{16} \oplus y_{21}$, $y_{36} \oplus y_{10} \oplus y_{14} \oplus y_{18} \oplus y_{22}$, $y_{41} \oplus y_9 \oplus y_{12} \oplus y_{20} \oplus y_{23}$, $y_{46} \oplus y_8 \oplus y_{15} \oplus y_{17} \oplus y_{24}$, $y_{51} \oplus y_7 \oplus y_{13} \oplus y_{19} \oplus y_{25}$. [Section 7.1.2 explains how to calculate majority functions efficiently. Notice that we can eliminate the last 10 bits if we only wish to correct up to two errors, and the last 20 if single-error correction is sufficient. See M. Y. Hsiao, D. C. Bossen, and R. T. Chien, *IBM J. Research and Development* **14** (1970), 390–394.]

25. By considering anagrams of {`l`, `e`, `a`, `s`, `t`} (see exercise 5–21), we're led to the square

```
stela
telas
elast ,
laste
astel
```

and the cyclic rotations of its rows. Here `telas` are Spanish fabrics; `elast` is a prefix meaning flexible; and `laste` is an imperative Chaucerian verb. (Of course just about every pronounceable combination of five letters has been used to spell or misspell something somewhere, at some point in history.)

26. "`every night, young video buffs catch rerun fever forty years after those great shows first aired`." [Robert Leighton, *GAMES* **16**, 6 (December 1992), 34, 47.]

27. $(0, 4, 163, 1756, 3834)$ for $k = (1, 2, 3, 4, 5)$; `mamma` and `esses` give a "full house."

28. Yes, 38 pairs altogether. The "most common" solution is `needs` (rank 180) and `offer` (rank 384). Only three cases differ consistently by $+1$ (`adder beefs`, `sheer tiffs`, `sneer toffs`). Other memorable examples are `ghost hints` and `strut rusts`. One word of the pair ends with the letter `s` except in four cases, such as `robed spade`. [See Leonard J. Gordon, *Word Ways* **23** (1990), 59–61.]

29. There are 18 palindromes, from `level` (rank 184) to `dewed` (rank 5688). Some of the 34 mirror pairs are '`devil lived`', '`knits stink`', '`smart trams`', '`faced decaf`'.

30. Among 105 such words in the SGB, `first`, `below`, `floor`, `begin`, `cells`, `empty`, and `hills` are the most common; `abbey` and `pssst` are lexicographically first and last. (If you don't like `pssst`, the next-to-last is `mossy`.) Only 37 words, from `mecca` to `zoned`, have their letters in *reverse* order; but they are, of course, `wrong` answers.

31. The middle word is the average of the other two, so the extreme words must be congruent mod 2; this observation reduces the number of dictionary lookups by a factor of about 32. There are 119 such triples in `WORDS`(5757), but only two in `WORDS`(2000): `marry`, `photo`, `solve`; `risky`, `tempo`, `vague`. [*Word Ways* **25** (1992), 13–15.]

32. The only reasonably common example seems to be `peopleless`.

33. `chief`, `fight`, `right`, `which`, `ouija`, `jokes`, `ankle`, `films`, `hymns`, `known`, `crops`, `pique`, `quart`, `first`, `first`, `study`, `mauve`, `vowel`, `waxes`, `proxy`, `crazy`, `pizza`. (The idea is to find the most common word in which x is followed by $(x+1) \bmod 26$, for $x = \mathtt{a}$ (0), $x = \mathtt{b}$ (1), …, $x = \mathtt{z}$ (25). We also minimize the intervening distance, thus preferring `bacon` to the more common word `black`. In the one case where no such word exists, `crazy` seems most rational. See *OMNI* **16**, 8 (May 1994), 94.)

34. The top two (and total number) in each category are: `pssst` and `pffft` (2), `schwa` and `schmo` (2), `threw` and `throw` (36), `three` and `spree` (5), `which` and `think` (709), `there` and `these` (234), `their` and `great` (291), `whooo` and `wheee` (3), `words` and `first` (628), `large` and `since` (376), `water` and `never` (1313), `value` and `radio` (84), `would` and `could` (460), `house` and `voice` (101), `quiet` and `queen` (25), `queue` only (1), `ahhhh` and `ankhs` (4), `angle` and `extra` (20), `other` and `after` (227), `agree` and `issue` (20), `along` and `using` (124), `above` and `alone` (92), `about` and `again` (58), `adieu` and `aquae` (2), `earth` and `eight` (16), `eagle` and `ounce` (8), `outer` and `eaten` (42), `eerie` and `audio` (4), (0), `ouija` and `aioli` (2), (0), (0); `years` and `every` are the most common of the 868 omitted words. [To fill the three holes, Internet usage suggests `ooops`, `ooooh`, and `ooooo`. See P. M. Cohen, *Word Ways* **10** (1977), 221–223.]

35. Consider the collection `WORDS`(n) for $n = 1, 2, \ldots, 5757$. The illustrated trie, rooted at `s`, first becomes possible when n reaches 978 (the rank of `stalk`). The next root letter to support such a trie is `c`, which acquires enough branching in its descendants when $n = 2503$ (the rank of `craze`). Subsequent breakthroughs occur when $n = 2730$ (`bulks`), 3999 (`ducky`), 4230 (`panty`), 4459 (`minis`), 4709 (`whooo`), 4782 (`lardy`), 4824 (`herem`), 4840 (`firma`), 4924 (`ridgy`), 5343 (`taxol`).

(A breakthrough occurs when a top-level trie acquires Horton–Strahler number 4; see exercise 7.2.1.6–124. Amusing sets of words, suggestive of a new kind of poetry, arise also when the branching is right-to-left instead of left-to-right: `black`, `slack`, `crack`, `track`, `click`, `slick`, `brick`, `trick`, `blank`, `plank`, `crank`, `drank`, `blink`, `clink`, `brink`, `drink`. In fact, right-to-left branching yields a complete *ternary* trie with 81 leaves: `males`, `sales`, `tales`, `files`, `miles`, `piles`, `holes`, …, `tests`, `costs`, `hosts`, `posts`.)

36. Denoting the elements of the cube by a_{ijk} for $1 \le i,j,k \le 5$, the symmetry condition is $a_{ijk} = a_{ikj} = a_{jik} = a_{jki} = a_{kij} = a_{kji}$. In general an $n \times n \times n$ cube has $3n^2$ words, obtained by fixing two coordinates and letting the third range from 1 to n; but the symmetry condition means that we need only $\binom{n+1}{2}$ words. Hence when $n = 5$ the number of necessary words is reduced from 75 to 15. [Jeff Grant was able to find 75 suitable words in the *Oxford English Dictionary*; see *Word Ways* **11** (1978), 156–157.]

Changing (`stove`, `event`) to (`store`, `erect`) or (`stole`, `elect`) gives two more.

37. The densest part of the graph, which we might call its “bare core,” contains the vertices named `bares` and `cores`, which each have degree 25.

38. `tears` $\to$ `raise` $\to$ `aisle` $\to$ `smile`; the second word might also be `reals`. [Going from `tears` to `smile` as in (11) was one of Lewis Carroll’s first five-letter examples. He would have been delighted to learn that the directed rule makes it more difficult to go from `smile` to `tears`, because *four* steps are needed in that direction.]

Internet
Cohen
Horton
Strahler
poetry
complete *ternary* trie
Grant
Dictionary
Carroll

Kempe
planar
Hamiltonian
girth
shrink any edge
puzzle
utilities
Dudeney
claw

39. Always spanning, never induced.

40. (a) 2^e, (b) 2^n, one for each subset of E or V.

41. (a) $n = 1$ and $n = 2$; P_0 is undefined. (b) $n = 0$ and $n = 3$.

42. G has $65/2$ edges (hence it doesn't exist).

43. Yes: The first three are isomorphic to Fig. 2(e). [The left-hand diagram is, in fact, identical to the earliest known appearance of the Petersen graph in print: See A. B. Kempe, *Philosophical Transactions* **177** (1886), 1–70, especially Fig. 13 in §59.] But the right-hand graph is definitely different; it is planar, Hamiltonian, and has girth 4.

44. Any automorphism must take a corner point into a corner point, because three distinct paths of length 2 can be found only between certain pairs of non-corner points. Therefore the graph has only the eight symmetries of C_4.

45. All edges of this graph connect vertices of the same row or adjacent rows. Therefore we can use the colors 0 and 2 alternately in even-numbered rows, 1 and 3 alternately in odd-numbered rows. The neighbors of `NV` form a 5-cycle, hence four colors are necessary.

46. (a) Every vertex has degree ≥ 2, and its neighbors have a well-defined cyclic order corresponding to the incoming lines. If $u \frac{\quad}{} v$ and $u \frac{\quad}{} w$, where v and w are cyclically consecutive neighbors of u, we must have $v \frac{\quad}{} w$. Thus all points in the vicinity of any vertex u belong to a unique triangular region.

(b) The formula holds when $n = 3$. If $n > 3$, shrink any edge to a point; this transformation removes one vertex and three edges. (If $u \frac{\quad}{} v$ shrinks, suppose it was part of the triangles $x \frac{\quad}{} u \frac{\quad}{} v \frac{\quad}{} x$ and $y \frac{\quad}{} u \frac{\quad}{} v \frac{\quad}{} y$. We lose vertex v and edges $\{x \frac{\quad}{} v, u \frac{\quad}{} v, y \frac{\quad}{} v\}$; all other edges of the form $w \frac{\quad}{} v$ become $w \frac{\quad}{} u$.)

47. A planar diagram would divide the plane into regions, with either 4 or 6 vertices in the boundary of each region (because $K_{3,3}$ has no odd cycles). If there are f_4 and f_6 of each kind, we must have $4f_4 + 6f_6 = 18$, since there are 9 edges; hence $(f_4, f_6) = (3, 1)$ or $(0, 3)$. We could also triangulate the graph by adding $f_4 + 3f_6$ more edges; but then it would have at least 15 edges, contradicting exercise 46.

[The fact that $K_{3,3}$ is nonplanar goes back to a puzzle about connecting three houses to three utilities (water, gas, and electricity), without crossing pipes. Its origin is unknown; H. E. Dudeney called it "ancient" in *Strand* **46** (1913), 110.]

48. If u, v, w are vertices and $u \frac{\quad}{} v$, we must have $d(w, u) \not\equiv d(w, v)$ (modulo 2); otherwise shortest paths from w to u and from w to v would yield an odd cycle. After w is colored 0, the procedure therefore assigns the color $d(w, v) \bmod 2$ to each new uncolored vertex v that is adjacent to a colored vertex u; and every vertex v with $d(w, v) < \infty$ is colored before a new w is chosen.

49. There are only three: K_4, $K_{3,3}$, and  (which is $\overline{C_6}$).

50. The graph must be connected, because the number of 3-colorings is divisible by 3^r when there are r components. It must also be contained in a complete bipartite graph $K_{m,n}$, which can be 3-colored in $3(2^m + 2^n - 2)$ ways. Deleting edges from $K_{m,n}$ does not decrease the number of colorings; hence $2^m + 2^n - 2 \leq 8$, and we have $\{m, n\} = \{1, 1\}$, $\{1, 2\}$, $\{1, 3\}$, or $\{2, 2\}$. So the only possibilities are the claw $K_{1,3}$ and the path P_4.

51. A 4-cycle $p_1 \frac{\quad}{} L_1 \frac{\quad}{} p_2 \frac{\quad}{} L_2 \frac{\quad}{} p_1$ would correspond to two distinct lines $\{L_1, L_2\}$ with two common points $\{p_1, p_2\}$, contradicting (ii). So the girth is at least 6.

If there's only one class of parallel lines, the girth is ∞; if there are two classes, it is 8. (See answer 22.) Otherwise we can find a 6-cycle by making a triangle from three lines that are chosen from different classes.

52. If the diameter is d and the girth is g, then $d \ge \lfloor g/2 \rfloor$, unless $g = \infty$.

53. `happy` (which is connected to `tears` and `sweat`, but not to `world`).

54. (a) It's a single, highly connected component. (Incidentally, this graph is the *line graph* of the bipartite graph in which one part corresponds to the initial letters $\{\mathtt{A}, \mathtt{C}, \mathtt{D}, \mathtt{F}, \mathtt{G}, \ldots, \mathtt{W}\}$ and the other to the final letters $\{\mathtt{A}, \mathtt{C}, \mathtt{D}, \mathtt{E}, \mathtt{H}, \ldots, \mathtt{Z}\}$.)

(b) Vertex `WY` is isolated. The other vertices with in-degree zero, namely `FL`, `GA`, `PA`, `UT`, `WA`, `WI`, and `WV`, form strong components by themselves; they all precede a giant strong component, which is followed by each of the remaining single-vertex strong components with out-degree zero: `AZ`, `DE`, `KY`, `ME`, `NE`, `NH`, `NJ`, `NY`, `OH`, `TX`.

(c) Now the strong component $\{\mathtt{GU}\}$ precedes $\{\mathtt{UT}\}$; `NH`, `OH`, `PA`, `WA`, `WI`, and `WV` join the giant strong component; $\{\mathtt{FM}\}$ precedes it; $\{\mathtt{AE}\}$ and $\{\mathtt{WY}\}$ follow it.

55. $\binom{N}{2} - \binom{n_1}{2} - \cdots - \binom{n_k}{2}$, where $N = n_1 + \cdots + n_k$.

56. True. Note that J_n is simple, but it doesn't correspond to any multigraph.

57. False, in the connected digraph $u \longrightarrow w \longleftarrow v$. (But u and v are in the same *strongly connected* component if and only if $d(u,v) < \infty$ and $d(v,u) < \infty$; see Section 2.3.4.2.)

58. Each component is a cycle whose order is at least (a) 3 (b) 1.

59. (a) By induction on n, we can use straight insertion sorting: Suppose $v_1 \longrightarrow \cdots \longrightarrow v_{n-1}$. Then either $v_n \longrightarrow v_1$ or $v_{n-1} \longrightarrow v_n$ or $v_{k-1} \longrightarrow v_n \longrightarrow v_k$, where k is minimum such that $v_n \longrightarrow v_k$. [L. Rédei, *Acta litterarum ac scientiarum* **7** (Szeged, 1934), 39–43.]

(b) 15: 01234, 02341, 02413, and their cyclic shifts. [The number of such oriented paths is always odd; see T. Szele, *Matematikai és Fizikai Lapok* **50** (1943), 223–256.]

(c) Yes. (By induction: If there's only one place to insert v_n as in part (a), the tournament is transitive.)

60. Set $A = \{x \mid u \longrightarrow x\}$, $B = \{x \mid x \longrightarrow v\}$, $C = \{x \mid v \longrightarrow x\}$. If $v \notin A$ and $A \cap B = \emptyset$ we have $|A| + |B| = |A \cup B| \le n-2$, because $u \notin A \cup B$ and $v \notin A \cup B$. But $|B| + |C| = n-1$; hence $|A| < |C|$. [H. G. Landau, *Bull. Math. Biophysics* **15** (1953), 148.]

61. $1 \longrightarrow 1$, $1 \longrightarrow 2$, $2 \longrightarrow 2$; then $A = \binom{1\,1}{0\,1}$ and $A^k = \binom{1\,k}{0\,1}$ for all integers k.

62. (a) Suppose the vertices are $\{1, \ldots, n\}$. Each of the $n!$ terms $a_{1p_1} \ldots a_{np_n}$ in the expansion of the permanent is the number of spanning permutation digraphs that have arcs $j \longrightarrow p_j$. (b) A similar argument shows that $\det A$ is the number of even spanning permutation digraphs minus the number of odd ones. [See F. Harary, *SIAM Review* **4** (1962), 202–210, where permutation digraphs are called "linear subgraphs."]

63. Let v be any vertex. If $g = 2t+1$, at least $d(d-1)^{k-1}$ vertices x satisfy $d(v,x) = k$, for $1 \le k < t$. If $g = 2t+2$ and v' is any neighbor of v, there also are at least $(d-1)^t$ vertices x for which $d(v,x) = t+1$ and $d(v',x) = t$.

64. To achieve the lower bound in answer 63, *every* vertex v must have degree d, and the d neighbors of v must all be adjacent to the remaining $d-1$ vertices. This graph is, in fact, $K_{d,d}$.

65. (a) By answer 63, G must be regular of degree d, and there must be exactly one path of length ≤ 2 between any two distinct vertices.

(b) We may take $\lambda_1 = d$, with $x_1 = (1 \ldots 1)^T$. All other eigenvectors satisfy $Jx_j = (0 \ldots 0)^T$; hence $\lambda_j^2 + \lambda_j = d-1$ for $1 < j \le N$.

(c) If $\lambda_2 = \cdots = \lambda_m = (-1+\sqrt{4d-3})/2$ and $\lambda_{m+1} = \cdots = \lambda_N = (-1-\sqrt{4d-3})/2$, we must have $m-1 = N-m$. With this value we find $\lambda_1 + \cdots + \lambda_N = d - d^2/2$.

line graph
bipartite graph
J_n
strongly connected
straight insertion sorting
Rédei
Szele
Landau
Harary
linear subgraphs
complete bigraph

(d) If $4d-3=s^2$ and m is as in (c), the eigenvalues sum to

$$\frac{s^2+3}{4}+(m-1)\frac{s-1}{2}-\Bigl(\frac{(s^2+3)^2}{16}+1-m\Bigr)\frac{s+1}{2},$$

which is 15/32 plus a multiple of s. Hence s must be a divisor of 15.

[These results are due to A. J. Hoffman and R. R. Singleton, *IBM J. Research and Development* **4** (1960), 497–504, who also proved that the graph for $d=7$ is unique.]

66. Denote the 50 vertices by $[a,b]$ and (a,b) for $0\le a,b<5$, and define three kinds of edges, using arithmetic mod 5:

$$[a,b] \mathrel{—} [a+1,b];\qquad (a,b)\mathrel{—}(a+2,b);\qquad (a,b)\mathrel{—}[a+bc,c]\quad\text{for } 0\le a,b,c<5.$$

[See W. G. Brown, *Canadian J. Math.* **19** (1967), 644–648; *J. London Math. Soc.* **42** (1967), 514–520. Without the edges of the first two kinds, the graph has girth 6 and corresponds to a geometric net as in exercise 51, using the orthogonal array in answer 19.]

67. Certain possibilities have been ruled out by Michael Aschbacher in *Journal of Algebra* **19** (1971), 538–540.

68. If G has s automorphisms, it has $n!/s$ adjacency matrices, because there are s permutation matrices P such that $P^{-}AP=A$.

69. First set `IDEG`$(v)\leftarrow 0$ for all vertices v. Then perform (31) for all v, also setting $u\leftarrow$ `TIP`(a) and `IDEG`$(u)\leftarrow$ `IDEG`$(u)+1$ in the second line of that mini-algorithm.

To do something "for all v" using the SGB format, first set $v\leftarrow$ `VERTICES`(g); then while $v<$ `VERTICES`(g) + `N`(g), do the operation and set $v\leftarrow v+1$.

70. Step B1 is performed once (but it takes $O(n)$ units of time). Steps (B2, B3, ..., B8) are performed respectively $(n+1,n,n,m+n,m,m,m,n)$ times, each with $O(1)$ cost.

71. Many choices are possible. Here we use 32-bit pointers, all relative to a symbolic address `Pool`, which lies in the `Data_Segment`. The following declarations provide one way to establish conventions for dealing with basic SGB data structures.

```
VSIZE IS 32 ;ASIZE IS 24             Node sizes
ARCS IS 0 ;COLOR IS 8 ;LINK IS 12    Offsets of vertex fields
TIP IS 0 ;NEXT IS 4                  Offsets of arc fields

arcs GREG Pool+ARCS ;color GREG Pool+COLOR ;link GREG Pool+LINK
tip GREG Pool+TIP ;next GREG Pool+NEXT
u GREG ;v GREG ;w GREG ;s GREG ;a GREG ;mone GREG -1

AlgB    BZ    n,Success            Exit if the graph is null.
        MUL   $0,n,VSIZE           B1. Initialize.
        ADDU  v,v0,$0              v ← v0 + n.
        SET   w,v0                 w ← v0.
1H      STT   mone,color,w         COLOR(w) ← −1.
        ADDU  w,w,VSIZE            w ← w + 1.
        CMP   $0,w,v
        PBNZ  $0,1B                Repeat until w = v.
0H      SUBU  w,w,VSIZE            w ← w − 1.
3H      LDT   $0,color,w           B3. Color w if necessary.
        PBNN  $0,2F                To B2 if COLOR(w) ≥ 0.
        STCO  0,link,w             COLOR(w) ← 0, LINK(w) ← Λ.
        SET   s,w                  s ← w.
4H      SET   u,s                  B4. Stack ⇒ u. Set u ← s.
```

Hoffman
Singleton
Brown
Aschbacher
automorphisms
permutation matrices
`VERTICES`(g)
`N`(g)

```
         LDTU    s,link,s        s ← LINK(s).
         LDT     $1,color,u
         NEG     $1,1,$1         $1 ← 1 − COLOR(u).
         LDTU    a,arcs,u        a ← ARCS(u).
5H       BZ      a,8F            B5. Done with u? To B8 if a = Λ.
5H       LDTU    v,tip,a         v ← TIP(a).
6H       LDT     $0,color,v      B6. Process v.
         CMP     $2,$0,$1        (Here the program is slightly clever)
         PBZ     $2,7F           To B7 if COLOR(v) = 1 − COLOR(u).
         BNN     $0,Failure      Fail if COLOR(v) = COLOR(u).
         STT     $1,color,v      COLOR(v) ← 1 − COLOR(u).
         STTU    s,link,v        LINK(v) ← s.
         SET     s,v             s ← v.
7H       LDTU    a,next,a        B7. Loop on a. Set a ← NEXT(a).
         PBNZ    a,5B            To B5 if a ≠ Λ.
8H       PBNZ    s,4B            B8. Stack nonempty? To B4 if s ≠ Λ.
2H       CMP     $0,w,v0         B2. Done?
         PBNZ    $0,0B           If w ≠ v0, decrease w and go to B3.
Success  LOC     @               (Successful termination) ▮
```

utility field
J_n

72. (a) This condition clearly remains invariant as vertices enter or leave the stack.

(b) Vertex v has been colored but not yet explored, because the neighbors of every explored vertex have the proper color.

(c) Just before setting $s \leftarrow v$ in step B6, set `PARENT`$(v) \leftarrow u$, where `PARENT` is a new utility field. And just before terminating unsuccessfully in that step, do the following: "Repeatedly output `NAME`(u) and set $u \leftarrow$ `PARENT`(u), until $u =$ `PARENT`(v); then output `NAME`(u) and `NAME`(v)."

73. K_{10}. (And *random_graph*$(10, 100, 0, 1, 1, 0, 0, 0, 0, 0)$ is J_{10}.)

74. `badness` has out-degree 22; no other vertices have out-degree > 20.

75. Let the parameters $(n_1, n_2, n_3, n_4, p, w, o)$ be respectively (a) $(n, 0, 0, 0, -1, 0, 0)$; (b) $(n, 0, 0, 0, 1, 0, 0)$; (c) $(n, 0, 0, 0, 1, 1, 0)$; (d) $(n, 0, 0, 0, -1, 0, 1)$; (e) $(n, 0, 0, 0, 1, 0, 1)$; (f) $(n, 0, 0, 0, 1, 1, 1)$; (g) $(m, n, 0, 0, 1, 0, 0)$; (h) $(m, n, 0, 0, 1, 2, 0)$; (i) $(m, n, 0, 0, 1, 3, 0)$; (j) $(m, n, 0, 0, -1, 0, 0)$; (k) $(m, n, 0, 0, 1, 3, 1)$; (l) $(n, 0, 0, 0, 2, 0, 0)$; (m) $(2, -n, 0, 0, 1, 0, 0)$.

76. Yes, for example from C_1 and C_2 in answer 75(c). (But no self-loops can occur when $p < 0$, because arcs $x \longrightarrow y = x + k\delta$ are generated for $k = 1, 2, \ldots$ until y is out of range or $y = x$.)

77. Suppose x and y are vertices with $d(x, y) > 2$. Thus $x \not\frac{\quad}{} y$; and if v is any other vertex we must have either $v \not\frac{\quad}{} x$ or $v \not\frac{\quad}{} y$. These facts yield a path of length at most 3 in $\overline{G}$ between any two vertices u and v.

78. (a) The number of edges, $\binom{n}{2}/2$, must be an integer. The smallest examples are K_0, K_1, P_4, C_5, and .

(b) If q is any odd number, we have $u \frac{\quad}{} v$ if and only if $\varphi^q(u) \not\frac{\quad}{} \varphi^q(v)$. Therefore φ^q cannot have two fixed points, nor can it contain a 2-cycle.

(c) Such a permutation of V also defines a permutation $\widehat{\varphi}$ of the edges of K_n, taking $\{u, v\} \mapsto \widehat{\varphi} = \{\varphi(u), \varphi(v)\}$, and it's easy to see that the cycle lengths of $\widehat{\varphi}$ are all multiples of 4. If $\widehat{\varphi}$ has t cycles, we obtain 2^t self-complementary graphs by painting the edges of each cycle with alternating colors.

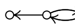
(d) In this case φ has a unique fixed point v, and $G' = G \setminus v$ is self-complementary. Suppose φ has r cycles in addition to (v); then $\widehat{\varphi}$ has r cycles involving the edges that touch vertex v, and there are 2^r ways to extend G' to a graph G.

[*References:* H. Sachs, *Publicationes Mathematicæ* **9** (Debrecen, 1962), 270–288; G. Ringel, *Archiv der Mathematik* **14** (1963), 354–358.]

79. Solution 1, by H. Sachs, with $\varphi = (1\,2\,\ldots\,4k)$: Let $u \mathrel{-\!\!-} v$ when $u > v > 0$ and $u + v \bmod 4 \le 1$; also $0 \mathrel{-\!\!-} v$ when $v \bmod 2 = 0$.

Solution 2, with $\varphi = (a_1\, b_1\, c_1\, d_1) \ldots (a_k\, b_k\, c_k\, d_k)$, where $a_j = 4j-3$, $b_j = 4j-2$, $c_j = 4j-1$, and $d_j = 4j$: Let $0 \mathrel{-\!\!-} b_j \mathrel{-\!\!-} a_j \mathrel{-\!\!-} c_j \mathrel{-\!\!-} d_j \mathrel{-\!\!-} 0$ for $1 \le j \le k$, and $a_i \mathrel{-\!\!-} a_j \mathrel{-\!\!-} b_i \mathrel{-\!\!-} d_j \mathrel{-\!\!-} c_i \mathrel{-\!\!-} c_j \mathrel{-\!\!-} d_i \mathrel{-\!\!-} b_j \mathrel{-\!\!-} a_i$, for $1 \le i < j \le k$.

80. (Solution by G. Ringel.) Let φ be as in answer 79, solution 2. Let E_0 be the $3k$ edges $b_j \mathrel{-\!\!-} a_j \mathrel{-\!\!-} c_j \mathrel{-\!\!-} d_j$ for $1 \le j \le k$; let E_1 be the $8\binom{k}{2}$ edges between $\{a_i, b_i, c_i, d_i\}$ and $\{b_j, d_j\}$ for $1 \le i < j \le k$; let E_2 be the $8\binom{k}{2}$ edges between $\{a_i, b_i, c_i, d_i\}$ and $\{a_j, c_j\}$ for $1 \le i < j \le k$. In case (a), $E_0 \cup E_1$ gives diameter 2; $E_0 \cup E_2$ gives diameter 3. Case (b) is similar, but we add $2k$ edges $b_j \mathrel{-\!\!-} 0 \mathrel{-\!\!-} d_j$ to E_1, $a_j \mathrel{-\!\!-} 0 \mathrel{-\!\!-} c_j$ to E_2.

81. $\vec{C_3}$, $\vec{K_3}$, $D =$ , and $D^T =$ . (The *converse* D^T of a digraph D is obtained by reversing the direction of its arcs. There are 16 nonisomorphic simple digraphs of order 3 without loops, 10 of which are self-converse, including $\vec{C_3}$ and $\vec{K_3}$.)

82. (a) True, by definition. (b) True: If every vertex has d neighbors, every edge $u \mathrel{-\!\!-} v$ has $d-1$ neighbors $u \mathrel{-\!\!-} w$ and $d-1$ neighbors $w \mathrel{-\!\!-} v$. (c) True: $\{a_i, b_j\}$ has $m+n-2$ neighbors, for $0 \le i < m$ and $0 \le j < n$. (d) False: $L(K_{1,1,2})$ has 5 vertices and 8 edges. (e) True. (f) True: The only nonadjacent edges are $\{0,1\} \not\mathrel{-\!\!-} \{2,3\}$, $\{0,2\} \not\mathrel{-\!\!-} \{1,3\}$, $\{0,3\} \not\mathrel{-\!\!-} \{1,2\}$. (g) True, for all $n > 0$. (h) False, unless G has no isolated vertices.

83. It is the Petersen graph. [A. Kowalewski, *Sitzungsberichte der Akademie der Wissenschaften in Wien*, Mathematisch-Nat. Klasse, Abteilung IIa, **126** (1917), 67–90.]

84. Yes: Let $\varphi(\{a_u, b_v\}) = \{a_{(u+v) \bmod 3}, b_{(u-v) \bmod 3}\}$ for $0 \le u, v < 3$.

85. Let the vertex degrees be $\{d_1, \ldots, d_n\}$. Then G has $\frac{1}{2}(d_1 + \cdots + d_n)$ edges, and $L(G)$ has $\frac{1}{2}(d_1(d_1-1) + \cdots + d_n(d_n-1))$. Thus G and $L(G)$ both have exactly n edges if and only if $(d_1-2)^2 + \cdots + (d_n-2)^2 = 0$. Consequently exercise 58 gives the answer. [See V. V. Menon, *Canadian Math. Bull.* **8** (1965), 7–15.]

86. If $G =$  then $\overline{G} =$  $= L(G)$.

87. (a) Yes, easily. [In fact, R. L. Brooks has proved that *every* connected graph with maximum vertex degree $d > 2$ is d-colorable, except for the complete graph K_{d+1}; see *Proc. Cambridge Phil. Soc.* **37** (1941), 194–197.]

(b) No. There's essentially only one way to 3-color the edges of the outer 5-cycle in Fig. 2(e); this forces a conflict on the inner 5-cycle. [Petersen proved this in 1898.]

88. One cycle doesn't use the center vertex, and there are $(n-1)(n-2)$ cycles that do (namely, one for every ordered pair of distinct vertices on the rim). We don't count C_0.

89. Both sides equal $\begin{pmatrix} A & O & O \\ O & B & O \\ O & O & C \end{pmatrix}$, $\begin{pmatrix} A & J & J \\ J & B & J \\ J & J & C \end{pmatrix}$, $\begin{pmatrix} A & J & J \\ O & B & J \\ O & O & C \end{pmatrix}$, $\begin{pmatrix} A & O & O \\ J & B & O \\ J & J & C \end{pmatrix}$, respectively.

90. K_4 and $\overline{K_4}$; $K_{1,1,2}$ and $\overline{K_{1,1,2}}$; $K_{2,2} = C_4$ and $\overline{K_{2,2}}$; $K_{1,3}$ and $\overline{K_{1,3}}$; $K_1 \oplus K_{1,2}$ and its complement; all graphs K_α are cographs by (47). Missing is $P_4 = \overline{P_4}$. (All connected subgraphs of a cograph have diameter ≤ 2; W_5 is a cograph, but not W_6.)

Sachs
Ringel
Sachs
Ringel
converse
simple digraphs
self-converse
isolated vertices
Petersen graph
Kowalewski
Menon
triangular grid
Brooks
complete graph
Petersen
diameter
wheel

91. (a) ; (b) ; (c) ; (d) ; (e) ; (f) ; (g) . (In general we have $K_2 \mathbin{\triangle} H = (K_2 \mathbin{\square} H) \cup (K_2 \otimes \overline{H})$, and $K_2 \circ H = H \mathbin{—} H$. Thus the coincidences $K_2 \mathbin{\triangle} H = K_2 \mathbin{\square} H$ and $K_2 \circ H = K_2 \boxtimes H$ occur if and only if H is a complete graph.)

Mnemonics: Our notations $G \mathbin{\square} H$ and $G \boxtimes H$ nicely match diagrams (a) and (c), as suggested by J. Nešetřil, *Lecture Notes in Comp. Sci.* **118** (1981), 94–102. His analogous recommendation to write $G \times H$ for (b) is also tempting; but it wasn't adopted here, because hundreds of authors have used $G \times H$ to denote $G \mathbin{\square} H$.

92. (a) ; (b) ; (c) ; (d) ; (e) .

93. $K_m \boxtimes K_n = K_m \circ K_n \cong K_{mn}$.

94. No; they're induced subgraphs of $K_{26} \mathbin{\square} K_{26} \mathbin{\square} K_{26} \mathbin{\square} K_{26} \mathbin{\square} K_{26}$.

95. (a) $d_u + d_v$. (b) $d_u d_v$. (c) $d_u d_v + d_u + d_v$. (d) $d_u(n - d_v) + (m - d_u)d_v$. (e) $d_u n + d_v$.

96. (a) $A \mathbin{\square} B = A \otimes I + I \otimes B$. (b) $A \boxtimes B = A \mathbin{\square} B + A \otimes B$. (c) $A \mathbin{\triangle} B = A \otimes J + J \otimes B - 2A \otimes B$. (d) $A \circ B = A \otimes J + I \otimes B$. (Formulas (a), (b), and (d) define graph products of arbitrary digraphs and multigraphs. Formula (c) is valid in general for simple digraphs; but negative entries can occur when A and B contain values > 1.)

Historical notes: The direct product of matrices is often called the Kronecker product, because K. Hensel [*Crelle* **105** (1889), 329–344] said he had heard it in Kronecker's lectures; however, Kronecker never actually published anything about it. Its first known appearance was in a paper by J. G. Zehfuss [*Zeitschrift für Math. und Physik* **3** (1858), 298–301], who proved that $\det(A \otimes B) = (\det A)^n (\det B)^m$ when $m = m'$ and $n = n'$. The basic formulas $(A \otimes B)^T = A^T \otimes B^T$, $(A \otimes B)(A' \otimes B') = AA' \otimes BB'$, and $(A \otimes B)^{-1} = A^{-1} \otimes B^{-1}$ are due to A. Hurwitz [*Math. Annalen* **45** (1894), 381–404].

97. Operations on adjacency matrices prove that $(G \oplus G') \mathbin{\square} H = (G \mathbin{\square} H) \oplus (G' \mathbin{\square} H)$; $(G \oplus G') \boxtimes H = (G \boxtimes H) \oplus (G' \boxtimes H)$; $(G \oplus G') \circ H = (G \circ H) \oplus (G' \circ H)$. Since $G \mathbin{\square} H \cong H \mathbin{\square} G$, $G \otimes H \cong H \otimes G$, and $G \boxtimes H \cong H \boxtimes G$, we also have right-distributive laws $G \mathbin{\square} (H \oplus H') \cong (G \mathbin{\square} H) \oplus (G \mathbin{\square} H')$; $G \otimes (H \oplus H') \cong (G \otimes H) \oplus (G \otimes H')$; $G \boxtimes (H \oplus H') \cong (G \boxtimes H) \oplus (G \boxtimes H')$. The lexicographic product satisfies $\overline{G \circ H} = \overline{G} \circ \overline{H}$; also $K_m \circ H = H \mathbin{—} \cdots \mathbin{—} H$, hence $K_m \circ \overline{K}_n = K_{n,\ldots,n}$. Furthermore $G \circ K_n = G \boxtimes K_n$; $K_m \otimes K_n = \overline{K_m \mathbin{\square} K_n} = L(K_{m,n})$.

98. There are kl components (because of the distributive laws in the previous exercise, and the facts that $G \mathbin{\square} H$ and $G \boxtimes H$ are connected when G and H are connected).

99. Every path from (u, v) to (u', v') in $G \mathbin{\square} H$ must use at least $d_G(u, u')$ "G-steps" and at least $d_H(v, v')$ "H-steps"; and that minimum is achievable. Similar reasoning shows that $d_{G \boxtimes H}((u, v), (u', v')) = \max(d_G(u, u'), d_H(v, v'))$.

100. If G and H are connected, and if each of them has at least two vertices, $G \otimes H$ is disconnected if and only if G and H are bipartite. The "if" part is easy; conversely, if there's an odd cycle in G, we can get from (u, v) to (u', v') as follows: First go to (u'', v'), where u'' is any vertex of G that happens to be expedient. Then walk an even number of steps in G from u'' to u', while alternating in H between v' and one of its neighbors. [P. M. Weichsel, *Proc. Amer. Math. Soc.* **13** (1962), 47–52.]

101. Choose vertices u and v with maximum degree. Then $d_u + d_v = d_u d_v$ by exercise 95; so either $G = H = K_1$, or $d_u = d_v = 2$. In the latter case, $G = P_m$ or C_m, and $H = P_n$ or C_n. But $G \mathbin{\square} H$ is connected, so G or H must be nonbipartite, say G. Then $G \mathbin{\square} H$ is nonbipartite, so H must also be nonbipartite; thus $G = C_m$ and $H = C_n$, with m and n both odd. The shortest cycle in $C_m \mathbin{\square} C_n$ has length $\min(m, n)$; in $C_m \otimes C_n$ it has length $\max(m, n)$; hence $m = n$. Conversely, if $n \geq 3$

Mnemonics
notations
Nešetřil
graph products
digraphs
multigraphs
simple digraphs
Kronecker product
Hensel
Zehfuss
Hurwitz
complementation
line graph
distributive laws
Weichsel

is odd, we have $C_n \,\square\, C_n \cong C_n \otimes C_n$, under the isomorphism that takes $(u,v) \mapsto ((u+v) \bmod n, (u-v) \bmod n)$ for $0 \le u, v < n$. [D. J. Miller, *Canadian J. Math.* **20** (1968), 1511–1521.]

Miller
planar
Erdős
Gallai
automorphism

102. $P_m \boxtimes P_n$. (It is planar only when $\min(m,n) \le 2$ or $m = n = 3$.)

103.

1	2	3	4	5	7
2	1	3	4	6	8
3	1	2	5	6	8
4	1	2	5	6	
5	3	4	1	7	
6	2	3	4		
7	5	1			
8	2	3			

1	2	3	4	5	6	7	8	9
2	1	3	4	6	8	9		
3	1	2	5	6	8	9		
4	1	2	5	7				
5	3	4	1	7				
6	2	3	1	7				
7	4	5	6	1				
8	2	3	1	9				
9	8	2	3	1				

104. Edges must be created in a somewhat circuitous order, to maintain the tableau shape. Variables r and i mark the starting and ending row in column t. For example, the second part of exercise 103 begins with $i \leftarrow 1$, $t \leftarrow 8$, $r \leftarrow 1$; then $9 \text{ — } 1$, $i \leftarrow 2$, $t \leftarrow 6$, $r \leftarrow 3$; then $9 \text{ — } 3$, $9 \text{ — } 2$, $i \leftarrow 4$, $t \leftarrow 4$, $r \leftarrow 8$; then $9 \text{ — } 8$.

105. Notice that $d_k \ge k$ if and only if $c_k \ge k$. When $d_k \ge k$ we have

$$c_1 + \cdots + c_k \;=\; k^2 + \min(k, d_{k+1}) + \min(k, d_{k+2}) + \cdots + \min(k, d_n);$$

therefore the condition $d_1 + \cdots + d_k \le c_1 + \cdots + c_k - k$ is equivalent to

$$d_1 + \cdots + d_k \le f(k), \quad \text{where } f(k) = k(k-1) + \min(k, d_{k+1}) + \cdots + \min(k, d_n). \quad (*)$$

If $k \ge s$ we have $f(k+1) - f(k) = 2k - d_{k+1} \ge d_{k+1}$; hence $(*)$ holds for $1 \le k \le n$ if and only if it holds for $1 \le k \le s$. Condition $(*)$ was discovered by P. Erdős and T. Gallai [*Matematikai Lapok* **11** (1960), 264–274]. It is obviously necessary, if we consider the edges between $\{1, \ldots, k\}$ and $\{k+1, \ldots, n\}$.

Let $a_k = d_1 + \cdots + d_k - c_1 - \cdots - c_k + k$, and suppose that $a_k > 0$ for some $k \le s$ after steps H3 and H4 have acted. Let A_j, C_j, D_j, N, and S be the numbers that correspond to a_j, c_j, d_j, n, and s *before* steps H3 and H4; thus $N = n + 1$, $D_j = d_j + (0 \text{ or } 1)$, etc. We want to prove that $A_K > 0$ for some $K \le S$.

Steps H3 and H4 have removed the bottommost q cells in column t, for some $t \ge S$, together with the rightmost cells in rows 1 through p, where $q + p = D_N$. Thus $A_j = a_j$ for $1 \le j \le p$; furthermore $A_j = a_j$ when $j \ge C_t$.

Let k be minimal with $a_k > 0$, and let $d_k = d$; notice that $c_k \le d$. If $d > t$ we have $k \le p$, hence $A_k = a_k > 0$. Therefore we may assume that $d = t - (0 \text{ or } 1)$, and $D_k = t$.

If $k < j \le C_t$ we have $d_j \ge D_j - 1 = t - 1 \ge d - 1 \ge c_k - 1 \ge c_j - 1$. Therefore $A_K = a_K \ge a_k$ when $K = C_t$; we may assume that $C_t > S$.

Now $D_S = D_{S+1} = t$, so $S = t$. Also $k = t$; otherwise $c_k \ge S + 1 > t \ge d$. Therefore $s = S$ and $d = t = c_t$. Further analysis shows that the only possibility with $A_t \le 0$ is $D_j = t + [j \ge t]$ for $1 \le j \le N = t + 2$. Algorithm H does indeed change this "good" sequence into a "bad" one; but $D_1 + \cdots + D_N = t^2 + 3t - 1$ is odd.

106. False in the trivial cases when $d \le 1$ and $n \ge d + 2$. Otherwise true: In fact, the first $n - 1$ edges generated in step H4 contain no cycles, so they form a spanning tree.

107. The permutation φ of exercise 78 takes a vertex of degree d into a vertex of degree $n-1-d$. And φ^2 is an automorphism that pairs up two vertices of equal degree, except for a possible fixed point of degree $(n-1)/2$.

(Conversely, a somewhat intricate extension of Algorithm H will construct a self-complementary graph from every graphical sequence that satisfies these conditions, provided that $d_{(n-1)/2} = (n-1)/2$ when n is odd. See C. R. J. Clapham and D. J. Kleitman, *J. Combinatorial Theory* **B20** (1976), 67–74.)

108. We may assume that $d_1^+ \ge \cdots \ge d_n^+$; the in-degrees d_k^- need not be in any particular order. Apply Algorithm H to the sequence $d_1 \ldots d_n = d_1^+ \ldots d_n^+$, but with the following changes: Step H2 becomes "[Done?] Terminate successfully if $d_1 = n = 0$; terminate unsuccessfully if $d_1 > n$." After setting i, t, and r in step H3, terminate unsuccessfully if $d_n^- > c_1$; otherwise do step H4 for $1 \le j \le d_n^-$, then set $n \leftarrow n-1$ and return to H2. In step H5, omit "$c_j \leftarrow c_j - 1$," and create the arc $k \longrightarrow n$ instead of the edge $k —— n$. An argument like Lemma M and Corollary H justifies this approach.

(Exercise 7.2.1.4–57 proves that such digraphs exist if and only if $d_1^- + \cdots + d_n^- = d_1^+ + \cdots + d_n^+$ and $d_1^- \ldots d_n^- = \{d_1', \ldots, d_n'\}$, where $d_1' \ge \cdots \ge d_n'$ and $d_1' \ldots d_n'$ is majorized by the conjugate partition $c_1 \ldots c_n = (d_1^+ \ldots d_n^+)^T$. The variant where loops $v \longrightarrow v$ are forbidden is harder; see D. R. Fulkerson, *Pacific J. Math.* **10** (1960), 831–836.)

109. It's the same as exercise 108, if we put $d_k^+ = d_k[k \le m]$ and $d_k^- = d_k[k > m]$.

110. There are p vertices of degree $d = d_1$ and q vertices of degree $d-1$, where $p+q = n$.

Case 1, $d = 2k+1$. Make $u —— v$ whenever $(u-v) \bmod n \in \{2, 3, \ldots, k+1, n-k-1, \ldots, n-3, n-2\}$; also add the $p/2$ edges $1 —— 2$, $3 —— 4$, …, $(p-1) —— p$.

Case 2, $d = 2k$. Make $u —— v$ whenever $(u-v) \bmod n \in \{2, 3, \ldots, k, n-k, \ldots, n-3, n-2\}$; also add the edges $1 —— 2$, …, $(q-1) —— q$, as well as the path or cycle $(q = 0?\ n{:}\ q) —— (q+1) —— \cdots —— (n-1) —— n$. [D. L. Wang and D. J. Kleitman, in *Networks* **3** (1973), 225–239, have proved that such graphs are highly connected.]

111. Suppose $N = n + n'$ and $V' = \{n+1, \ldots, N\}$. We want to construct $e_k = d - d_k$ edges between k and V', and additional edges within V', so that each vertex of V' has degree d. Let $s = e_1 + \cdots + e_n$. This task is possible only if (i) $n' \ge \max(e_1, \ldots, e_n)$; (ii) $n'd \ge s$; (iii) $n'd \le s + n'(n'-1)$; and (iv) $(n+n')d$ is even.

Such edges do exist whenever n' satisfies (i)–(iv): First, s suitable edges between V and V' can be created by cyclically choosing endpoints $(n{+}1, n{+}2, \ldots, n{+}n', n{+}1, \ldots)$, because of (i). This process assigns either $\lfloor s/n' \rfloor$ or $\lceil s/n' \rceil$ edges to each vertex of V'; we have $\lceil s/n' \rceil \le d$ by (ii), and $d - \lfloor s/n' \rfloor \le n' - 1$ by (iii). Therefore the additional edges needed inside V' are constructible by exercise 110 and (iv).

The choice $n' = n$ always works. Conversely, if $G = K_n(V) \setminus \{1 —— 2\}$, condition (iii) requires $n' \ge n$ when $n \ge 4$. [P. Erdős and P. Kelly, *AMM* **70** (1963), 1074–1075.]

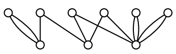
112. The uniquely best triangle in the *miles* data is

$$\texttt{Saint Louis, MO} \overset{748}{——} \texttt{Toronto, ON} \overset{746}{——} \texttt{Winston-Salem, NC} \overset{748}{——} \texttt{Saint Louis, MO.}$$

113. By Murphy's Law, it has n rows and m columns; so it's $n \times m$, not $m \times n$.

114. A loop in a multigraph is an edge $\{a, a\}$ with repeated vertices, and a multigraph is a 2-uniform hypergraph. Thus we should allow the incidence matrix of a general hypergraph to have entries greater than 1 when an edge contains a vertex more than once. (A pedant would probably call this a "multihypergraph.") With these considerations in mind, the incidence matrix and bipartite graph corresponding to (26) are

$$\begin{pmatrix} 210000 \\ 011100 \\ 001122 \end{pmatrix};$$



115. The element in row e and column f of B^TB is $\sum_v b_{ve} b_{vf}$; so B^TB is $2I$ plus the adjacency matrix of $L(G)$. Similarly, BB^T is D plus the adjacency matrix of G, where D is the diagonal matrix with d_{vv} = degree of v. (See exercises 2.3.4.2–18, 19, and 20.)

Clapham
Kleitman
majorized
conjugate partition
Fulkerson
Wang
Kleitman
vertex connectivity
Erdős
Kelly
Murphy's Law
loop
multihypergraph
adjacency matrix
line graph
degree

116. $\overline{K_{m,n}^{(r)}} = K_m^{(r)} \oplus K_n^{(r)}$, generalizing (38), for all $r \geq 1$.

117. The nonisomorphic multisets of singleton edges for $m = 4$ and $V = \{0, 1, 2\}$ are $\{\{0\},\{0\},\{0\},\{0\}\}$, $\{\{0\},\{0\},\{0\},\{1\}\}$, $\{\{0\},\{0\},\{1\},\{1\}\}$, and $\{\{0\},\{0\},\{1\},\{2\}\}$. The answer in general is the number of partitions of m into at most n parts, namely $\left|{m+n \atop n}\right|$, using the notation explained in Section 7.2.1.4. (Of course, there's little reason to think of partitions as 1-uniform hypergraphs, except when answering strange exercises.)

118. Let d be the sum of the vertex degrees. The corresponding bipartite graph is a forest with $m + n$ vertices, d edges, and p components. Hence $d = m + n - p$, by Theorem 2.3.4.1A.

119. Then there's an additional edge, containing all seven vertices.

120. We could say that (hyper)arcs are arbitrary sequences of vertices, or sequences of distinct vertices. But most authors seem to define hyperarcs to be $A \longrightarrow v$, where A is an unordered set of vertices. When the best definition is found, it will probably be the one that has the most important practical applications.

121. $\chi(H) = |F| - \alpha(I(H)^T)$ is the size of a minimum cover of V by sets of F.

122. (a) One can verify that there are just seven 3-element covers, namely the vertices of an edge; so there are seven 4-element independent sets, namely the complements of an edge. We can't two-color the hypergraph, because one color would need to be used 4 times and the other three colors would be an edge. (Hypergraph (56) is essentially the projective plane with seven points and seven lines.)

(b) Since we're dualizing, let's call the vertices and edges of the Petersen graph "points" and "lines"; then the vertices and edges of the dual are lines and points, respectively. Color red the five lines that join an outer point to an inner point. The other ten lines are independent (they don't contain all three of the lines touching any point); so they can be colored green. No set of eleven lines can be independent, because no four lines can touch all ten points. (Thus the Petersen dual is a bipartite hypergraph, in spite of the fact that it contains cycles of length 5.)

123. They correspond to $n \times n$ latin squares, whose entries are the vertex colors.

124. Four colors easily suffice. If it were 3-colorable, there must be four vertices of each color, since no five vertices are independent. Then two opposite corners must have the same color, and a contradiction arises quickly.

125. The Chvátal graph is the smallest such graph with $g = 4$. G. Brinkmann found the smallest with $g = 5$: It has 21 vertices a_j, b_j, c_j for $0 \leq j < 7$, with edges $a_j \mathbin{—} a_{j+2}$, $a_j \mathbin{—} b_j$, $a_j \mathbin{—} b_{j+1}$, $b_j \mathbin{—} c_j$, $b_j \mathbin{—} c_{j+2}$, $c_j \mathbin{—} c_{j+3}$ and subscripts mod 7. M. Meringer showed that there must be at least 35 vertices if $g > 5$. B. Grünbaum conjectured that g can be arbitrarily large; but no further constructions are known. [See *AMM* **77** (1970), 1088–1092; *Graph Theory Notes of New York* **32** (1997), 40–41.]

126. When m and n are even, both C_m and C_n are bipartite, and 4-coloring is easy. Otherwise a 4-coloring is impossible. When $m = n = 3$, a 9-coloring is optimum by exercise 93. When $m = 3$ and $n = 4$ or 5, at most two vertices are independent; it's easy to find an optimum 6- or 8-coloring. Otherwise we obtain a 5-coloring by painting vertex (j, k) with $(a_j + 2b_k) \bmod 5$, where periodic sequences $\langle a_j \rangle$ and $\langle b_k \rangle$ exist with period lengths m and n, respectively, such that $a_j - a_{j+1} \equiv \pm 1$ and $b_k - b_{k+1} \equiv \pm 1$ for all j and k. [K. Vesztergombi, *Acta Cybernetica* **4** (1978), 207–212.]

127. (a) The result is true when $n = 1$. Otherwise let $H = G \setminus v$, where v is any vertex. Then $\overline{H} = \overline{G} \setminus v$, and we have $\chi(H) + \chi(\overline{H}) \leq n$ by induction. Clearly $\chi(G) \leq \chi(H) + 1$

partitions
bipartite graph
hyperarcs
projective plane
bipartite hypergraph
latin squares
Chvátal graph
Brinkmann
Meringer
Grünbaum
Vesztergombi

and $\chi(\overline{G}) \le \chi(\overline{H}) + 1$; so there's no problem unless equality holds in all three cases. But that can't happen; it implies that $\chi(H) \le d$ and $\chi(\overline{H}) \le n-1-d$, where d is the degree of v in G. [E. A. Nordhaus and J. W. Gaddum, *AMM* **63** (1956), 175–177.]

To get equality, let $G = K_a \oplus \overline{K_b}$, where $ab > 0$ and $a+b = n$. Then we have $\overline{G} = \overline{K_a} \mathbin{—} K_b$, $\chi(G) = a$, and $\chi(\overline{G}) = b+1$. [All graphs for which equality holds have been found by H.-J. Finck, *Wiss. Zeit. der Tech. Hochschule Ilmenau* **12** (1966), 243–246.]

(b) A k-coloring of G has at least $\lceil n/k \rceil$ vertices of some color; those vertices form a clique in $\overline{G}$. Hence $\chi(G)\chi(\overline{G}) \ge \chi(G)\lceil n/\chi(G) \rceil \ge n$. Equality holds when $G = K_n$.

(From (a) and (b) we deduce that $\chi(G)+\chi(\overline{G}) \ge 2\sqrt{n}$ and $\chi(G)\chi(\overline{G}) \le \frac{1}{4}(n+1)^2$.)

128. $\chi(G \mathbin{\square} H) = \max(\chi(G), \chi(H))$. This many colors is clearly necessary. And if the functions $a(u)$ and $b(v)$ color G and H with the colors $\{0, 1, \ldots, k-1\}$, we can color $G \mathbin{\square} H$ with $c(u, v) = (a(u) + b(v)) \bmod k$.

129. A complete row or column (16 cases); a complete diagonal of length 4 or more (18 cases); a 5-cell pattern $\{(x, y), (x{-}a, y{-}a), (x{-}a, y{+}a), (x{+}a, y{-}a), (x{+}a, y{+}a)\}$ for $a \in \{1, 2, 3\}$ $(36+16+4$ cases); a 5-cell pattern $\{(x, y), (x{-}a, y), (x{+}a, y), (x, y{-}a), (x, y{+}a)\}$ for $a \in \{1, 2, 3\}$ $(36 + 16 + 4$ cases); a pattern containing four of those five cells, when the fifth lies off the board $(24 + 32 + 24$ cases); or a 4-cell pattern $\{(x, y), (x{+}a, y), (x, y{+}a), (x{+}a, y{+}a)\}$ for $a \in \{1, 3, 5, 7\}$ $(49 + 25 + 9 + 1$ cases). Altogether 310 maximal cliques, with respectively $(168, 116, 4, 4, 18)$ of size $(4, 5, 6, 7, 8)$.

130. If graph G has p maximal cliques and graph H has q, then the join $G \mathbin{—} H$ has pq, because the cliques of $G \mathbin{—} H$ are simply the unions of cliques from G and H. Furthermore, the empty graph $\overline{K_n}$ has n maximal cliques (namely its singleton sets).

Thus the complete k-partite graph with part sizes $\{n_1, \ldots, n_k\}$, being the join of empty graphs of those sizes, has $n_1 \ldots n_k$ maximal cliques.

131. Assume that $n > 1$. In a complete k-partite graph, the number $n_1 \ldots n_k$ is maximized when each part has size 3, except perhaps for one or two parts of size 2. (See exercise 7.2.1.4–68(a).) So we must prove that $N(n)$ cannot be larger than this in *any* graph.

Let $m(v)$ be the number of maximal cliques that contain vertex v. If $u \not\mathbin{—} v$ and $m(u) \le m(v)$, construct the graph G' that is like G except that u is now adjacent to all the neighbors of v instead of to its former neighbors. Every maximal clique U in either graph belongs to one of three classes:

i) $u \in U$; there are $m(u)$ of these in G and $m(v)$ of them in G'.
ii) $v \in U$; there are $m(v)$ of these in G and also in G'.
iii) $u \notin U$ and $v \notin U$; such maximal cliques in G are also maximal in G'.

Therefore G' has at least as many maximal cliques as G. And we can obtain a complete k-partite graph by appropriately repeating the process.

[This argument, due to Paul Erdős, was presented by J. W. Moon and L. Moser in *Israel J. Math.* **3** (1965), 23–25.]

132. The strong product of cliques in G and H is a clique in $G \boxtimes H$, by exercise 93; hence $\omega(G \boxtimes H) \ge \omega(G)\omega(H) = \chi(G)\chi(H)$. On the other hand, colorings $a(u)$ and $b(v)$ of G and H lead to the coloring $c(u, v) = (a(u), b(v))$ of $G \boxtimes H$; hence $\chi(G \boxtimes H) \le \chi(G)\chi(H)$. And $\omega(G \boxtimes H) \le \chi(G \boxtimes H)$.

133. (a) 24; (b) 60; (c) 3; (d) 6; (e) 6; (f) 4; (g) 5; (h) 4; (i) $K_2 \boxtimes C_{12}$; (j) 18; (k) 12. (l) Yes, of degree 5. (m) No. [Can it be drawn with fewer than 12 crossings?] (n) Yes; in fact, it is 4-connected (see Section 7.4.1). (o) Yes; we consider *every* graph to be directed, with two arcs for each edge. (p) Of course not. (q) Yes, easily.

Nordhaus
Gaddum
Finck
join
Erdős
Moon
Moser
crossings
connectivity

[The musical graph represents simple modulations between key signatures. It appears on page 73 of *Graphs* by R. J. Wilson and J. J. Watkins (1990).]

Wilson
Watkins
strong product
Holton
Sheehan
raman

134. By rotating and/or swapping the inner and outer vertices, we can find an automorphism that takes any vertex into C. If C is fixed, we can interchange the inner and outer vertices of any subset of the remaining 11 pairs, and/or do a left-right reflection. Therefore there are $24 \times 2^{11} \times 2 = 98{,}304$ automorphisms altogether.

135. Let $\omega = e^{2\pi i/6}$, and define the matrices $Q = (q_{ij})$, $S = (s_{ij})$, where $q_{ij} = [j = (i+1) \bmod 12]$ and $s_{ij} = \omega^{ij}$, for $0 \le i, j < 12$. By exercise 96(b), the adjacency matrix of the musical graph $K_2 \boxtimes C_{12}$ is $A = \binom{1\,1}{1\,1} \otimes (I+Q+Q^-) - I$. Let T be the matrix $\binom{1\;\;\,1}{1\,-1} \otimes S$; then $T^- AT$ is a diagonal matrix D whose first 12 entries are $1 + 4\cos\frac{j\pi}{6}$ for $0 \le j < 12$, and whose other 12 entries are -1. Therefore $A^{2m} = TD^m T^-$, and it follows that the number of $2m$-step walks from C to $(\mathrm{C}, \mathrm{G}, \mathrm{D}, \mathrm{A}, \mathrm{E}, \mathrm{B}, \mathrm{F}^\sharp)$ respectively is

$$\begin{aligned}
\mathrm{C}_m &= \tfrac{1}{24}(25^m + 2(13+4\sqrt{3})^m + 3^{2m+1} + 2(13-4\sqrt{3})^m + 16);\\
\mathrm{G}_m &= \tfrac{1}{24}(25^m + \sqrt{3}(13+4\sqrt{3})^m - \sqrt{3}(13-4\sqrt{3})^m - 1);\\
\mathrm{D}_m &= \tfrac{1}{24}(25^m + (13+4\sqrt{3})^m + (13-4\sqrt{3})^m - 3);\\
\mathrm{A}_m &= \tfrac{1}{24}(25^m - 3^{2m+1} + 2);\\
\mathrm{E}_m &= \tfrac{1}{24}(25^m - (13+4\sqrt{3})^m - (13-4\sqrt{3})^m + 1);\\
\mathrm{B}_m &= \tfrac{1}{24}(25^m - \sqrt{3}(13+4\sqrt{3})^m + \sqrt{3}(13-4\sqrt{3})^m - 1);\\
\mathrm{F}^\sharp_m &= \tfrac{1}{24}(25^m - 2(13+4\sqrt{3})^m + 3^{2m+1} - 2(13-4\sqrt{3})^m);
\end{aligned}$$

also $\mathrm{a}_m = \mathrm{C}_m - 1$, $\mathrm{d}_m = \mathrm{F}_m = \mathrm{e}_m = \mathrm{G}_m$, etc. In particular, $(\mathrm{C}_6, \mathrm{G}_6, \mathrm{D}_6, \mathrm{A}_6, \mathrm{E}_6, \mathrm{B}_6, \mathrm{F}^\sharp_6) =$ (15462617, 14689116, 12784356, 10106096, 7560696, 5655936, 5015296), so the desired probability is $15462617/5^{12} \approx 6.33\%$. As $m \to \infty$, the probabilities are all $\frac{1}{24} + O(0.8^m)$.

136. No. Only two Cayley graphs of order 10 are cubic, namely $K_2 \square C_5$ (whose vertices can be written $\{e, \alpha, \alpha^2, \alpha^3, \alpha^4, \beta, \beta\alpha, \beta\alpha^2, \beta\alpha^3, \beta\alpha^4\}$ where $\alpha^5 = \beta^2 = (\alpha\beta)^2 = e$) and the graph with vertices $\{0, 1, \ldots, 9\}$ and arcs $v \to (v \pm 1) \bmod 10$, $v \to (v+5) \bmod 10$. [See D. A. Holton and J. Sheehan, *The Petersen Graph* (1993), exercise 9.10. Incidentally, the SGB graphs $raman(p, q, t, 0)$ are Cayley graphs.]

137. Let $[x, y]$ denote the label of (x, y); we want $[x, y] = [x+a, y+b] = [x+c, y+d]$ for all x and y. If A is the matrix $\binom{a\;b}{c\;d}$, the operation of adding t times the bottom row of A to the top row changes A to the matrix $A' = \binom{1\;t}{0\;1}A = \binom{a'\;b'}{c'\;d'}$, where $a' = a + tc$, $b' = b + td$, $c' = c$, $d' = d$. The new condition $[x, y] = [x + a', y + b'] = [x + c', y + d']$ is equivalent to the old; and $\gcd(a', b', c', d') = \gcd(a, b, c, d)$. Similarly we can premultiply A by $\binom{1\;0}{t\;1}$ without really changing the problem.

We can also operate on columns, replacing A by $A'' = A\binom{1\;t}{0\;1} = \binom{a''\;b''}{c''\;d''}$, where $a'' = a$, $b'' = ta+b$, $c'' = c$, $d'' = tc+d$. This operation does alter the problem, but only slightly: If we find a labeling that satisfies $[\![x, y]\!] = [\![x + a'', y + b'']\!] = [\![x + c'', y + d'']\!]$ for all x and y, then we'll have $[x, y] = [x+a, y+b] = [x+c, x+d]$ if $[x, y] = [\![x, y+tx]\!]$. Similarly we can postmultiply A by $\binom{1\;0}{t\;1}$; the problem remains almost the same.

A series of such row and column operations will reduce A to the simple form $UAV = \binom{1\;0}{0\;n}$, where U and V are integer matrices with $\det U = \det V = 1$. And if we have $V = \binom{\alpha\;\beta}{\gamma\;\delta}$, a labeling for the reduced problem that satisfies the simple conditions $[\![x, y]\!] = [\![x+1, y]\!] = [\![x, y+n]\!]$ will provide a solution to the original labeling problem if we define $[x, y] = [\![\alpha x + \gamma y, \beta x + \delta y]\!]$.

Finally, the reduced labeling problem is easy: We let $[\![x,y]\!] = y \bmod n$. Thus the desired answer is to set $p=\beta$, $q=\delta$.

138. Proceeding as before, but with a $k\times k$ matrix A, row and column operations will reduce the problem to a diagonal matrix UAV. The diagonal entries $(d_1,\ldots,d_k)$ are characterized by the condition that $d_1\ldots d_j$ is the greatest common divisor of the determinants of all $j\times j$ submatrices of A. [This is "Smith normal form"; see H. J. S. Smith, *Philosophical Transactions* **151** (1861), 293–326, §14.] If the labeling $[\![x]\!]$ satisfies the reduced problem, the original problem is satisfied by $[x]=[\![xV]\!]$. The number of elements in the generalized torus is $n=\det A = d_1\ldots d_k$.

The reduced problem has a simple solution as before if $d_1=\cdots=d_{k-1}=1$. But in general the reduced labeling will be an r-dimensional ordinary torus of dimensions $(d_{k-r+1},\ldots,d_k)$, where $d_{k-r+1}>d_{k-r}=1$. (Here $d_0=1$; we might have $r=k$.)

In the requested example, we find $d_1=1$, $d_2=2$, $d_3=10$, $n=20$; indeed,

$$UAV = \begin{pmatrix}1&-2&0\\0&1&-1\\-1&-1&4\end{pmatrix}\begin{pmatrix}3&1&1\\1&3&1\\1&1&3\end{pmatrix}\begin{pmatrix}1&5&6\\0&1&1\\0&0&1\end{pmatrix} = \begin{pmatrix}1&0&0\\0&2&0\\0&0&10\end{pmatrix}.$$

Each point (x,y,z) now receives a two-dimensional label $(u,v) = ((5x+y)\bmod 2,$ $(6x+y+z)\bmod 10)$. The six neighbors of (u,v) are $((u\pm1)\bmod 2, v)$, $((u\pm1)\bmod 2,$ $(v\pm1)\bmod 10)$, $(u,(v\pm1)\bmod 10)$. It's a multigraph, since the first two neighbors are identical; but it's not the same as the multigraph $C_2\boxtimes C_{10}$, which has degree 8.

[Generalized toruses are essentially the Cayley graphs of Abelian groups; see exercise 136. They have been proposed as convenient interconnection networks, in which case it is desirable to minimize the diameter when k and n are given. See C. K. Wong and D. Coppersmith, *JACM* **21** (1974), 392–402; C. M. Fiduccia, R. W. Forcade, and J. S. Zito, *SIAM J. Discrete Math.* **11** (1998), 157–167.]

139. (This exercise helps clarify the distinction between labeled graphs G, in which the vertices have definite names, and unlabeled graphs H such as those in Fig. 2.) If N_H is the number of labeled graphs on $\{1,2,\ldots,h\}$ that are isomorphic to H, and if U is any h-element subset of V, the probability that $G\,|\,U$ is isomorphic to H is $N_H/2^{h(h-1)/2}$. Therefore the answer is $\binom{n}{h}N_H/2^{h(h-1)/2}$. We need only figure out the value of N_H, which is: (a) 1; (b) $h!/2$; (c) $(h-1)!/2$; (d) $h!/a$, where H has a automorphisms.

140. (a) $\#(K_3{:}W_n)=n-1$ and $\#(P_3{:}W_n)=\binom{n-1}{2}$ for $n\ge5$; also $\#(\overline{K_3}{:}W_8)=7$.

(b) G is proportional if and only if $\#(K_3{:}G)=\#(\overline{K_3}{:}G)=\frac18\binom{n}{3}$ and $\#(P_3{:}G)=\#(\overline{P}_3{:}G)=\frac38\binom{n}{3}$. If G has e edges, we have $(n-2)e = 3\#(K_3{:}G)+2\#(P_3{:}G)+\#(\overline{P}_3{:}G)$, because every pair of vertices appears in $n-2$ induced subgraphs. If G has degree sequence $d_1\ldots d_n$, we have $d_1+\cdots+d_n=2e$, $\binom{d_1}{2}+\cdots+\binom{d_n}{2}=3\#(K_3{:}G)+\#(P_3{:}G)$, and $d_1(n-1-d_1)+\cdots+d_n(n-1-d_n)=2\#(P_3{:}G)+2\#(\overline{P}_3{:}G)$. Therefore a proportional graph satisfies $(*)$ — unless $n=2$. (The exercise should have excluded that case.)

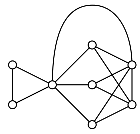
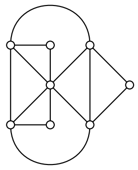
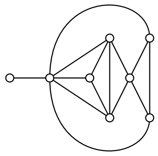
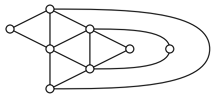
Conversely, if G satisfies $(*)$ and has the correct $\#(K_3{:}G)$, it also has the correct $\#(P_3{:}G)$, $\#(\overline{P}_3{:}G)$, and $\#(\overline{K_3}{:}G)$.

[*References:* S. Janson and J. Kratochvíl, *Random Structures & Algorithms* **2** (1991), 209–224. In *J. Combinatorial Theory* **B47** (1989), 125–145, A. D. Barbour, M. Karoński, and A. Ruciński had shown that the variance of $\#(H{:}G)$ is proportional to either n^{2h-2}, n^{2h-3}, or n^{2h-4}, where the first case occurs when H does not have $\frac12\binom{h}{2}$ edges, and the third case occurs when H is a proportional graph.]

141. Only 8 degree sequences $d_1\ldots d_8$ satisfy $(*)$: 73333333 (1/2), 65433322 (26/64), 64444222 (2/10), 64443331 (8/22), 55543222 (8/20), 55533331 (2/10), 55444321 (26/64),

diagonal matrix
determinants
Smith normal form
multigraph
Cayley graphs
Abelian groups
Wong
Coppersmith
Fiduccia
Forcade
Zito
labeled graphs
unlabeled graphs
isomorphic
automorphisms
Janson
Kratochvíl
Barbour
Karoński
Ruciński
variance

and 44444440 (1/2). Each degree sequence is shown here with statistics (N_1/N), where N nonisomorphic graphs have that sequence and N_1 of them are proportional. The last three cases are complements of the first three. No graph of order 8 is both proportional and self-complementary. Maximally symmetric examples of the first five cases are W_8,

, , , and .

142. The hint follows as in the previous answer; $(n-3)\#(\overline{K}_3{:}G)$ and $(n-3)\#(P_3{:}G)$ can also be expressed in terms of four-vertex counts. Furthermore, a graph with e edges has $\binom{e}{2} = \#(P_3 \subseteq G) + \#(K_2 \oplus K_2 \subseteq G)$, because any two edges form either P_3 or $K_2 \oplus K_2$; in this formula, $\#(P_3 \subseteq G)$ counts not-necessarily-induced subgraphs.

We have $\#(P_3 \subseteq G) = \#(P_3{:}G) + 3\#(K_3{:}G)$, and a similar formula expresses $\#(K_2 \oplus K_2 \subseteq G)$ in terms of induced counts. Thus an extraproportional graph must be proportional and satisfy $e = \frac{1}{2}\binom{n}{2}$, $\#(P_3 \subseteq G) = \frac{3}{4}\binom{n}{3}$, $\#(K_2 \oplus K_2 \subseteq G) = \frac{3}{4}\binom{n}{4}$. But these values contradict the formula for $\binom{e}{2}$.

143. Consider the graph whose vertices are the rows of A, and whose edges $u \longrightarrow v$ signify that rows u and v agree except in one column, j. Label such an edge j.

If the graph contains a cycle, delete any edge of the cycle, and repeat the process until no cycles remain. Notice that the label on every deleted edge appears elsewhere in its cycle; hence the deletions don't affect the set of edge labels. But we're left with fewer than $m \le n$ edges, by Theorem 2.3.4.1A; so there are fewer than n different labels. [See J. A. Bondy, *J. Combinatorial Theory* **B12** (1972), 201–202.]

144. Let G be the graph on vertices $\{1,\ldots,m\}$, with edges $i \longrightarrow j$ if and only if $* \ne x_{il} \ne x_{jl} \ne *$ for some l. This graph is k-colorable if and only if there is a completion with at most k distinct rows. Conversely, if G is a graph on vertices $\{1,\ldots,n\}$, with adjacency matrix A, the $n \times n$ matrix $X = A + *(J - I - A)$ has the property that $i \longrightarrow j$ if and only if $* \ne x_{il} \ne x_{jl} \ne *$ for some l. [See M. Sauerhoff and I. Wegener, *IEEE Trans.* **CAD-15** (1996), 1435–1437.]

145. Set $c \leftarrow 0$ and repeat the following operations for $1 \le j \le n$: If $c = 0$, set $x \leftarrow a_j$ and $c \leftarrow 1$; otherwise if $x = a_j$, set $c \leftarrow c + 1$; otherwise set $c \leftarrow c - 1$. Then x is the answer. The idea is to keep track of a possible majority element x, which occurs c times in nondiscarded elements; we discard a_j and one x whenever finding $x \ne a_j$. [See *Automated Reasoning* (Kluwer, 1991), 105–117. Extensions to find all elements that occur more than n/k times, in $O(n \log k)$ steps, have been discussed by J. Misra and D. Gries, *Science of Computer Programming* **2** (1982), 143–152.]

self-complementary
Bondy
adjacency matrix
Sauerhoff
Wegener
Misra
Gries

INDEX AND GLOSSARY

When an index entry refers to a page containing a relevant exercise, see also the *answer* to that exercise for further information. An answer page is not indexed here unless it refers to a topic not included in the statement of the exercise.