

Section Handout #7—Graphs

The purpose of this section is to let you work with the most important graph algorithms. Each of these questions uses the definitions of the **Graph** class and the **Node** and **Arc** structures as they appear in Chapter 18.

1. Coding depth-first search

Write a function

```
bool pathExists(Node *n1, Node *n2);
```

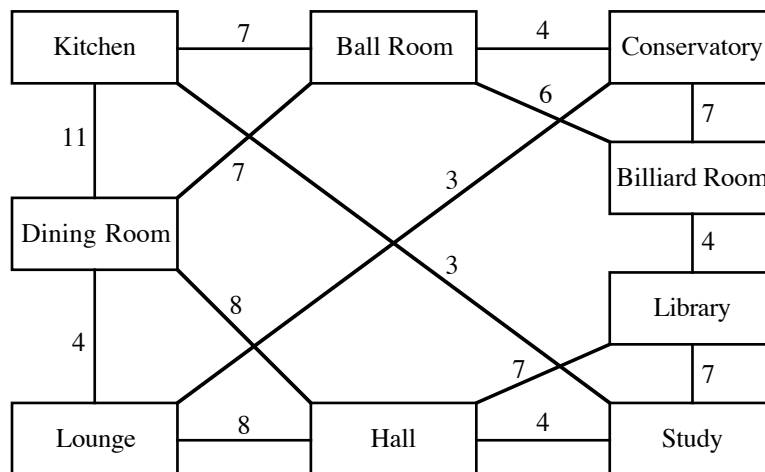
that returns **true** if there is a path in the graph between the nodes **n1** and **n2**. In this version of the exercise, implement this function by using depth-first search to traverse the graph from **n1**. If you encounter **n2** along the way, then a path exists.

2. Coding breadth-first search

Rewrite the **pathExists** function with so that the algorithm uses a breadth-first search instead.

3. Understanding graph algorithms

Those of you who have played Clue (or Cluedo if you come from outside North America) will recognize the following undirected graph, which shows the connections between the various rooms on the game board:



The numbers on the various arcs show the distance (measured in spaces on the board) between pairs of rooms. For example, the distance from the Hall to the Lounge is 8 steps, and the distance from the Ball Room to the Billiard Room is 6 steps. In this problem, the “secret passages” that connect the rooms at the corners of the board (the Kitchen-Study and Lounge-Conservatory arcs) are arbitrarily assumed to have distance 3.

- 3a)** Indicate the order of traversal for a depth-first search starting at the Lounge. Assume that iteration over a set chooses nodes in alphabetical order, as it does in the **Graph** class. Thus, the first step in the depth-first search will be to the Conservatory, rather than to the Dining Room or Hall, both of which come later in the alphabet.
- 3b)** Indicate the order of traversal for a breadth-first search starting at the Kitchen. As before, assume that nodes in any set are processed in alphabetical order.
- 3c)** Trace the operation of Dijkstra's algorithm to find the minimum path from the Lounge to the Library.
- 3d)** Trace the operation of Kruskal's algorithm to find the minimum spanning tree for the Clue board.