

Problem Set #2

Due: Friday, March 6 by 5:00 P.M. Problem sets are pencil-and-paper exercises and may be submitted in section, in lecture, or to the box outside my office (Gates 202). If you turn in your answers on this page, be sure to fill out the following information:

Name: _____ Section Leader: _____

Problem 1: Binary search trees

Diagram the binary search tree that results from adding—in order—the English words for the first ten counting numbers:

"one", "two", "three", "four", "five", "six", "seven", "eight", "nine", "ten"

to an empty binary search tree. As in the section problem, you should use the following definition for the nodes of the BST:

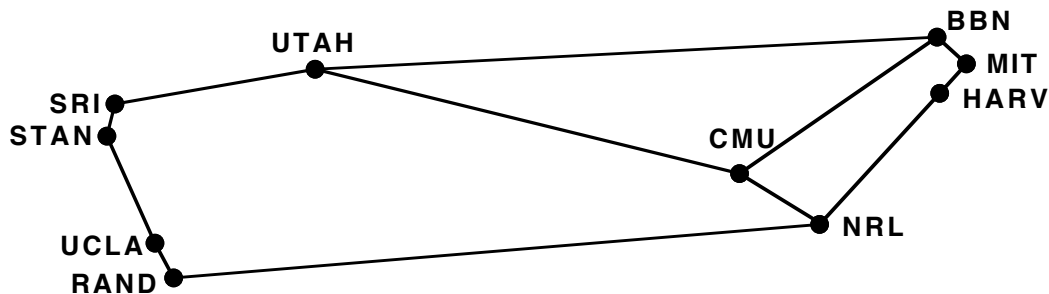
```
struct BSTNode {  
    string key;  
    BSTNode *left, *right;  
};
```

Your diagram can be very simple and need show only the value of the key in each node and a line connecting that node to its left and right subtrees. In the problem, you can omit empty subtrees entirely.

Once you have completed the diagram, determine the height of the resulting tree and which nodes, if any, are out of balance.

Problem 2: Graph traversals

The first ten nodes on the ARPANET (the forerunner to the modern Internet) were connected to form the following graph:

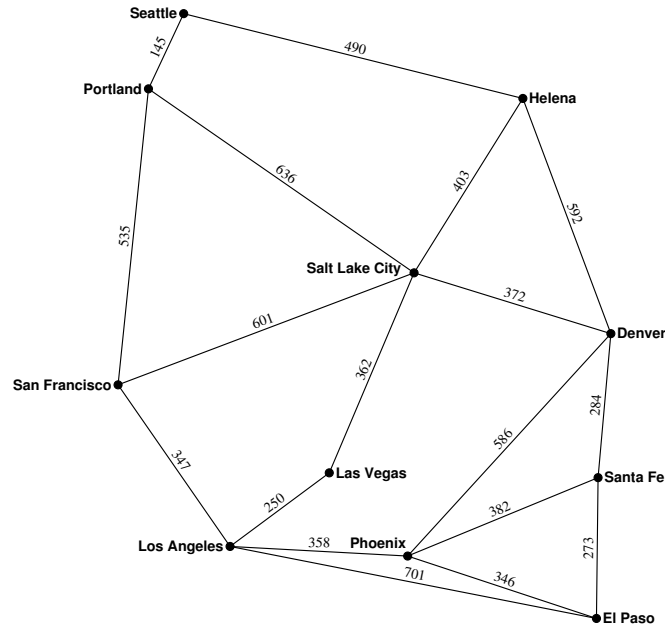


Assuming that node sets are processed in alphabetical order, in what order do the nodes get visited if you perform a recursive depth-first search from the Stanford node (**STAN**)?

In what order do the nodes get visited if you perform a breadth-first search starting at Carnegie Mellon (**CMU**)?

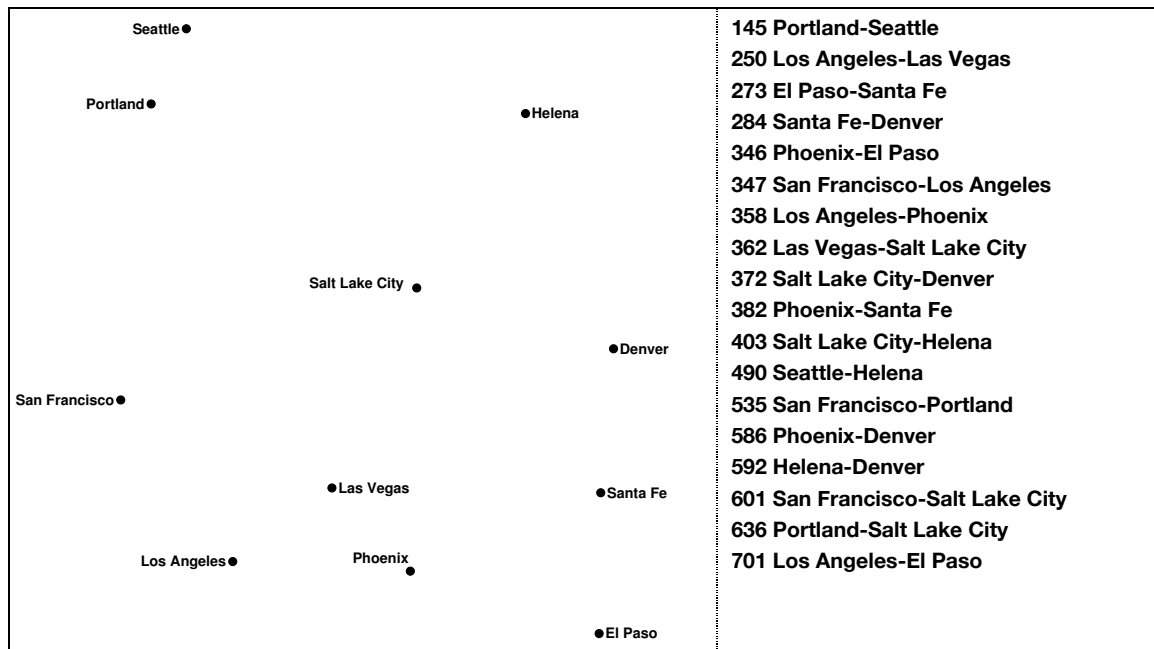
3. Minimum spanning trees

The money that Leland Stanford used to found our university came largely from his holdings in the Central Pacific Railroad, so it seems appropriate to use a railroad graph as an example, which in this case is adapted from the game *Ticket to Ride*:



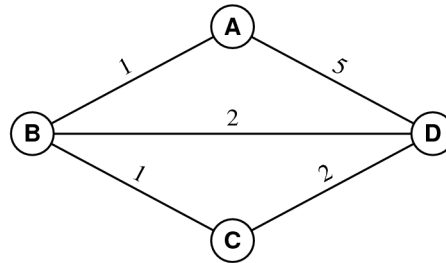
Apply Kruskal's algorithm to this graph to find the minimum spanning tree of this graph. Show your answer by adding the arcs in the spanning tree to the empty city graph below. To make your job easier, we've given you the list of arcs in the column at the right, sorted in increasing order by distance as the algorithm requires.

Answer to Problem 3



4. Dijkstra's algorithm

Apply Dijkstra's algorithm to find the shortest path from node **A** to node **D** in the following graph:



Show each individual step in the algorithm by showing the contents of the priority queue each time a node is enqueued or dequeued.

5. Expression trees

Go through the parsing of the expression

$$y = 2 + (x - 3) * 5$$

and list all the calls to **readT** and **readE** in the order in which they occur. Draw a diagram showing the structure of the final expression tree.

Once you have created the expression tree, evaluate the expression in a context in which **x** has the value 11. Show the value of each call to **eval** in the order in which those calls return.