COSC 2123/1285 Algorithms and Analysis
Tutorial 3
Brute Force Algorithmic Paradigm

Objective

Students who complete this tutorial should:

- Understand and apply selection and bubble sort.
- Understand and apply exhaustive search.
- Understand and apply BFS and DFS graph traversals.

Questions

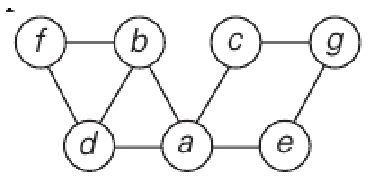
3.1.8 Sort the list "`E, X, A, M, P, L, E`" in alphabetical order (ascending order) by selection sort.

Answer: See lecture notes, but scan through list, find smallest letter, put into first position. Then continue with rest of list until all sorted.

3.1.11 Sort the list "`E, X, A, M, P, L, E`" in alphabetical order (ascending order) by bubble sort.

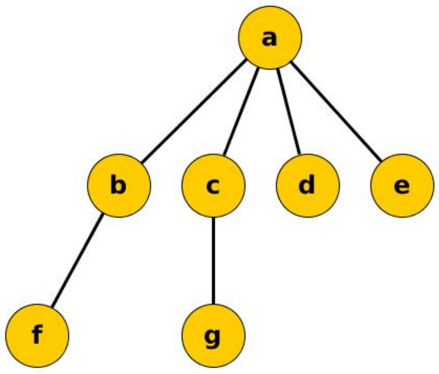
Answer: See lecture notes, but scan through list, swapping adjacent letters if they are out of place. Then largest letter should be at last position. Continue with bubbling second largest letter to second last position, then repeat unitl all letters are sorted.

3.5.4 Traverse the following graph by breadth-first search (BFS) and construct the corresponding breadth-first search tree. Start the traversal at vertex a and resolve ties by the vertex alphabetical order.



Answer: BFS traversal: a, b, c, d, e, f, g

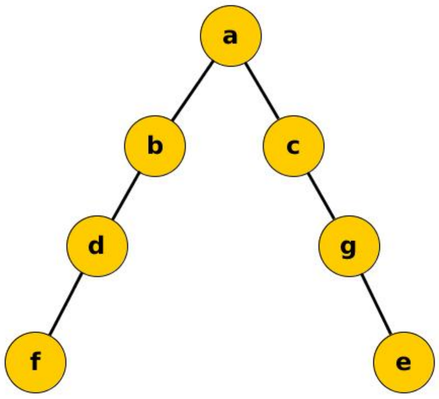
BFS Tree:



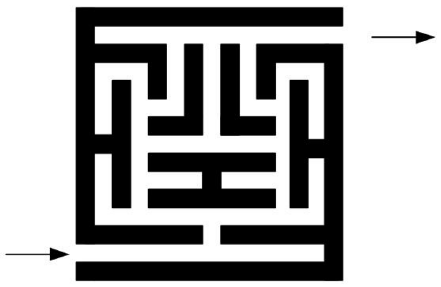
3.5.1b Repeat question 3.5.4, but now using depth-first search (DFS) traversal. Also start the traversal at vertex a and resolve ties by the vertex alphabetical order.

Answer: DFS traversal: a, b, d, f, c, g, e

DFS Tree:



3.5.1 One can model a maze by having a vertex for a starting point, a finishing point, dead ends, and all the points in the maze where more than one path can be taken, and then connecting the vertices according to the paths in the maze.



a Construct such a graph for the maze.

b Which traversal – DFS or BFS – would you use if you found yourself in a maze and why?

Answer:

1. For each fork or dead-end in the maze, we represent that as a vertex. If there is a direct path between forks, or a fork and dead-end, then add an edge.

2. DFS is much more convenient for going through a maze than BFS. When DFS moves to a next vertex, it is connected to a current vertex by an edge (i.e., "close nearby" in the physical maze), which is not generally the case for BFS. In fact, DFS can be considered a generalisation of an ancient right-hand rule for maze traversal: go through the maze in such a way so that your right hand is always touching a wall.

3.4.6 Consider the **partition** problem: given n positive integers, partition them into two disjoint subsets with the same sum of their elements. (Of course, the problem does not always have a solution.) Design an exhaustive-search algorithm for this problem. Try to minimize the number of subsets the algorithm needs to generate.

Answer: Start by computing the sum $\mathcal{S}$ of the numbers given. If $\mathcal{S}$ is odd, stop because the problem doesn't have a solution. If $\mathcal{S}$ is even, generate all solutions with subsets that have $n/2$ or less elements. Check if any of solutions have subsets that sum to $\mathcal{S}/2$ (a solution).