CS 337: Algorithms: Design & Practice

Lab 6: Simple Graph Algorithms

Secret Identity: Two plays that have been on Broadway within the past 100 years.

The "Death of a Salesman" Puzzle – Traveling sales from Washington to Washington, DC

In this lab we will learn how to represent graphs and implement some simple graph algorithms to solve a problem based (loosely) on the play "Death of a Salesman" by Arthur Miller. First the puzzle:

You need to travel, by land, from the state of Washington to Washington, DC. You are only allowed to travel through states whose names begin with any of the letters in the word in the first initial and last name of the titular character in the play — Willie Loman. That is, you can travel through states beginning with W, L, O, M, A, N and no others. So, **A**rkansas, **M**innesota, **W**yoming, etc are OK but not **C**alifornia, or **T**exas, or **U**tah, etc. Is it possible? If so, what would be the path?

Graphs: In class this week we reviewed the basics of graphs/networks: terminology, properties, and some basic algorithms (See Chapters 5 & 6 from Cormen). You can solve the puzzle above by modeling it as a graph search problem:

Given a graph of all the states in the United States and their neighbors, you can model the problem as a graph search problem where Washington is the start state and Washington, DC the goal state. A database of states and neighbors is available in a text file on Unix

/home/gtowell/Public/CS337/USStates.csv

Here are some of the entries in it:

```
Colorado, Arizona, Kansas, Nebraska, New Mexico, Oklahoma, Utah, Wyoming
Connecticut, Massachusetts, New York, Rhode Island
District of Columbia, Maryland, Virginia
Delaware, Maryland, New Jersey, Pennsylvania
Florida, Alabama, Georgia
Georgia, Alabama, Florida, North Carolina, South Carolina, Tennessee
Hawaii,
Idaho, Montana, Nevada, Oregon, Utah, Washington, Wyoming
Illinois, Indiana, Iowa, Michigan, Kentucky, Missouri, Wisconsin
```

Each line represents a state, followed by all its neighbors.

Task#1: Create a graph ADT (Abstract Data Type) to input and store the above graph. The program should print out the state with the most neighbors. And then, in your main program, use the graph to answer some simple queries:

```
Enter a state: Florida
Florida has the following neighbors: Alabama, Georgia

Enter a state: Idaho
Idaho has the following neighbors: Montana, Nevada, Oregon,
Utah, Washington, Wyoming
```

Enter a state: Hawaii
Hawaii has no neighbors.

Task#2: Solve the "Death Of A Salesman" Puzzle. Your program should first say "Yes" or "No" whether there is a way to go from Washington to Washington, DC. If the answer is "Yes" it should print out the names of the states the path goes through to get from Washington to Washington, DC (District of Columbia). That is:

No. There is no way to get from Washington to District of Columbia.

Or

Yes. To get from Washington to District of Columbia, march as follows:
Washington, Oregon, Nevada, …, District of Columbia

Think about the best possible implementation of the graph ADT, the algorithm(s) you will use, and how they will be implemented. Your implementation should make judicious use of already implemented data types and structures in the library of the language of your choice.

Your program will be graded based not just on correctness, but also on choice of algorithm(s), programming style, and use of proper abstractions.

Extra Credit: Suppose we change the destination to Pennsylvania (it does not begin with a 'W'). For each state in the US, can you print out whether it would be possible to travel from there to Pennsylvania (i.e. to Bryn Mawr College) and the path, if possible. The constraints of traveling through "salesman" states remain the same except for the start and end states (e.g. starting in Georgia is it possible to travel to Pennsylvania by traveling only through "salesman" states?).

What to submit:

Create a written report containing the following:

1. A discussion on this lab describing the graph data structure you used. Your report should have a lot of figures, much of the text should be descriptions of those figures. In your narrative, name the graph search algorithm employed, and the specific data structures/libraries you used. This should be followed by a printout of sample runs from Task#1 (as shown in the handout).
2. A printout of the output from Task#2.
3. A discussion summarizing the results from Extra Credit if you did it.
4. The Appendix should include a printout of your program from Task#2. A printout from Extra Credit if you did it.