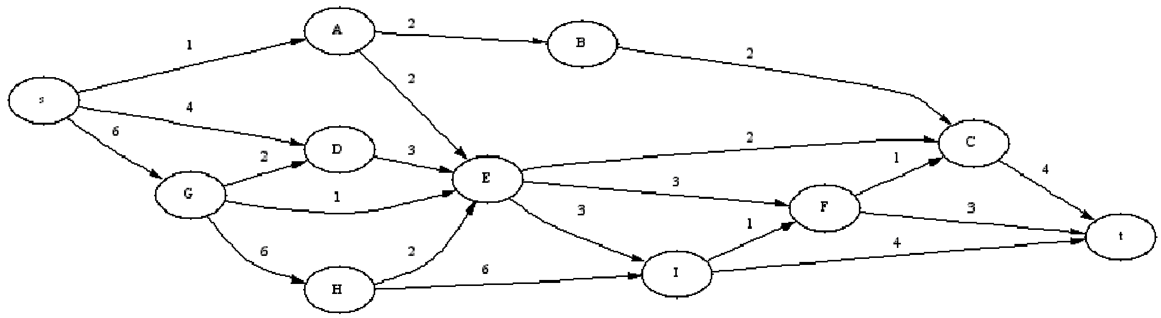
Homework 5

Data Structures and Algorithms in C++
Shlomo Hershkop
Department of Computer Science
Columbia University
Summer 2006

Due Aug 10 at 11pm

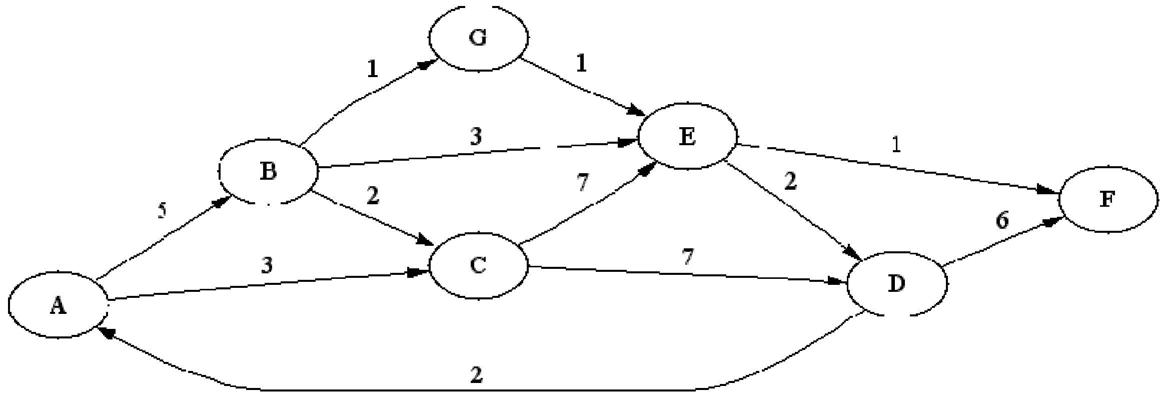
Theory (100 points)

1) Find a topological sort, DFS expansion, and BFS expansion for this graph:



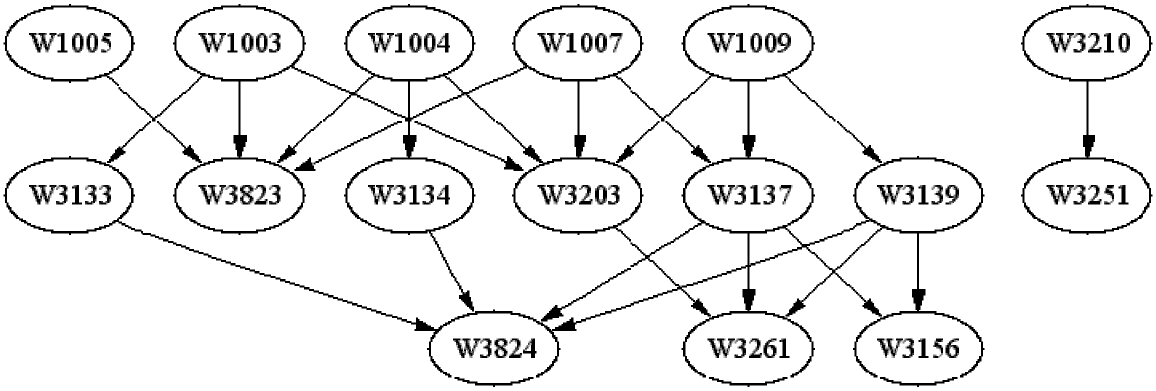
2) Give an example where Dijkstra's algorithm gives the wrong answer in the presence of a negative edge but no negative-cost cycle.

3) Execute Dijkstra's algorithm from node a in the following graph:



4) A bipartite graph G = (V,E) is a graph such that V can be partitioned into two subsets V1 and V2 and no edge has vertices in the same subset. Give a linear algorithm to determine whether a graph is bipartite.

5) For the following graph,



a. Draw this as an adjacency matrix
b. Draw it as an adjacency list
c. Describe a short function (psuede code) to convert the input of this graph into an adjacency list, assuming the input is a text file with each line having a direct edge as (x,v) which means a directed edge from x to v.

6) Show that the algorithm for finding a cycle in the graph using BFS or DFS must be O(n)

7) For a directed graph with n vertices, what is the largest number of distinct solutions for the topological sort of the graph ?

8) The notion of a minimum spanning tree is applicable to a connected weighted graph. Do we have to check for a graph's connectivity before applying Prim's algorithm or can the algorithm do it by itself?

9) Will either Kruskal's algorithm or Prim's algorithm work on a graph with negative edges weights?

10) Under what circumstances is the minimum spanning tree found by the algorithm unique ?

Programming Section (100 Points)

There are many websites with mapping software, and many travel software, but not many which combine the two. You will be coding the backbone of such a system.

You will see a file called worldcities.txt which contains the GPS coordinates of 9117 cities around the world. The format of the file is :
N (number of cities)
city comma state
Lat
Long
etc

Note: the names might have spaces. In addition I will provide a short test file called fict100.txt which has a sample of 100 fake cities. (the format is the same). Do not hard code the number of cities, but rather read in the number on the first line.

You program should do the following:

1) take in an argument as to which file to read
   example: *yourprogram* fict100.txt
2) It should read in all the cities and sort them by state, so that you can quickly find all cities by state
3) You will need to keep GPS information in the graph node data structure since you will be using it for calculations.
4) You will need to present the user with the following menu items
   a. Search for state
   b. Search for city
   c. Set current city
   d. Show current city
   e. Find n closest
   f. Find shortest path
   g. quit

   Here is an explanation of the above menu choices

   a. Search for state
      will return all cities associated with the specific state. For example if the user enters ‘a’, prompt for state, and if they enter “New York” show all cities in new york…..along with some ID number (which you create)

   b. Search for city
      will return the ID number of the specific city if its in the system

c. Set current city
will take an ID and remember it as the current city

d. Show current city
will print the currently set city

e. Find n closest
find the n closest cities from the current city, if the current has not been set choose a random city, n is a number provided by the user.
for calculating gps distances see:
http://www.meridianworlddata.com/Distance-Calculation.asp

f. Find shortest path
will take an ID and calculate all shortest hops from the current city to the destination city, finding the shortest path.

g. Quit
hmmm…this should quit

Have fun and please specify in your README and comments which algorithms you are implementing and their expected runtimes….note if you see yourself coding a n^{10} algorithm…find a better one