CSCI 335
Fifth assignment

Total: 100 points
Extra Credit: 15 points

Due 5/15/12
Please, follow the blackboard instructions on writing and submitting programming assignments
We will not debug your assignment. It should run correctly to receive credit.

Graph Algorithms [100 points]

You will read a directed graph from a text file. Below is an example:

Graph1.txt

```
5
0  1 0.2 3 10.1 4 0.5 -1
1  0 1.5 -1
2  1 100.0 3 50.2 -1
3  -1
4   1 10.5 2 13.9 -1
```

The first line is the number of vertices N (5 in this example). Each vertex is represented by an integer from 0 to N-1 (0 through 4 in this example).
For each vertex you have a set of the adjacent vertices along with their positive weights. This set terminates with -1 (note that we don't use negative weights here).
For instance, in the above example, vertex 0 is connected to vertex 1 (edge weight 0.2), to vertex 3 (edge weight 10.1) and to vertex 4 (edge weight 0.5).

You can use your own graphs for your experiments (plot them, and create their text files by hand). We will also give you a number of graph files to experiment with.

A) [10 points] Represent a graph using an adjacency list. Read the vertices and edges from the text file.

B) [30 points] Implement the Dijkstra algorithm for weighted shortest path starting from a particular vertex. Use a priority-queue implementation. Prompt the user to specify the vertex you want to start from. After the algorithm terminates

   a. Display the number of operations performed.
   b. Prompt the user to enter a specific vertex. Then your program should print out the cost of the shortest path to that vertex and the sequence of vertices along the path.

C) [20 points] Treat the weights as flows. Prompt the user to enter two vertices A and B. Compute the maximum flow from A to B (this requires a small modification to the Dijkstra algorithm). Note that this is not the complete network flow algorithm, just the part that computes the augmenting path. Print the augmented path generated along with the maximum flow.

D) [20 points] Using (C) implement the network flow algorithm. Prompt the user to specify a source S and destination vertex D. Then print the maximum flow for the graph, and save the flow graph into a text file (using the format shown at the beginning of this document).

E) [20 points] Implement the DFS algorithm. Use it in order to determine the articulation points of an undirected graph.

EXTRA CREDIT:

Implement and test an all-source shortest path algorithm using dynamic programming (15 points).