

# Csci 335 Programming Project 3

---

*Due Tuesday, December 22 @ NOON*

## Graph Representation

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  $N - 1$ .

For each vertex you have a list of the adjacent vertices with positive edge weights. Each list terminates with -1 to indicate the end of the line. 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).

Represent a graph using an adjacency list. The list of neighbors can be implemented using a vector or a linked list. Read the vertices and edges from a text file.

## Dijkstra's Algorithm

Implement Dijkstra's Algorithm using a priority queue. You don't have to use your implementation of priority queue. You may use the `std::priority_queue` container from STL.

Write a program that runs as follows:

```
findPaths <GRAPH_FILE> <STARTING_VERTEX>
```

This program should use Dijkstra's Algorithm to find the shortest paths from a given starting vertex to all vertices in the graph file. The program should then continuously prompt the user to specify a target vertex. It will then print the sequence of vertices along the shortest path from the starting vertex to the target vertex as well as the total cost of the path.

As always, your code should be able to compile via "make all".