Csci 335 Assignment 3

Due Monday December 22nd

Graph Representation

Create a representation for a weighted undirected graph using an adjacency list. Read section 9.1.1 in the book for suggestions on how to do this.

To make your implementation work for undirected graphs then whenever you add directed edge (u, v) you should also add the directed edge (v, u).

Prim's Algorithm

Implement Prim's Algorithm for finding the Minimum Spanning Tree of a weighted graph using the mutable priority queue from the previous assignment.

Alternatively you may implement the version that uses a regular priority queue. Leave a comment if you choose to do this.

Minimum Edit Distance

Implement the dynamic programming algorithm for finding the minimum edit distance between two strings of characters.

You may also use the C++ implementation here:
http://en.wikibooks.org/wiki/Algorithm_Implementation/Strings/Levenshtein_distance#C.2B.2B

Application

You will be given three files, `allnames.txt`, `girlnames.txt` and `boynames.txt`, containing the top 1000 baby names for boys and girls from 2012 as provided by the Social Security Administration website. Your task is to create a system for finding similar baby names using the above two algorithms.

First, construct a graph of names weighted by the minimum edit distance between names. For every pair of names add a weighted edge between those names if the weight is ≤ 4. Even though we compute weights for every possible pair of edges, this will be a relatively sparse graph. To make sure that it is sparse count and print out the total number of vertices and edges created.

Next, given a name and a depth threshold value as input, run Prim's algorithm starting on the given name to find the Minimum Spanning Tree. Using the output of Prim's algorithm, traverse the Minimum Spanning Tree in a depth-first order up to the given depth value threshold.

There are several ways to implement this last step. One way I would recommend is to use the output of Prim's algorithm to construct a new graph. Then in this graph perform a depth-first search up to the given depth limit.

Another way is to keep track of the unweighted distance of each vertex from the starting vertex in Prim's algorithm and reject adding any vertex beyond the depth threshold. The algorithm will stop earlier and you simply need to perform a depth-first traversal on the resulting tree.

For each name in the tree, print out the name as well as the weight of the edge to its parent.

Name your program `findNames` and it should run for example as follows (I will supply the marynames.txt file for testing purposes):

```
./findNames marynames.txt
Building graph.
Total number of nodes: 8
Total number of edges: 26

Enter a name and depth limit (Type "exit" or Ctrl + C to stop):
>>> Mary 3

Names similar to Mary:

Mary (0)
      Maria (2)
            Mariah (1)
            Marian (1)
                  Maryanne (3)
            Marla (1)
            Marisa (1)
      Marley (2)

Enter a name and depth limit (Type "exit" or Ctrl + C to stop):
>>>
```

Working in a Group of Two

For this assignment you may work individually or in groups of **two**. If you choose to work in a group of two then please email me telling me who is in your group. Cc your partner on this email.