CS215 Spring 2010 Program 5: aMazing

In Program4 you were able to read a data file and create a variety of data structures. One of which formed a sort of 'maze' linking various Sector objects together in a particular fashion.

These Sector objects contained a property named Size, but no size was specified in any of the input. Now you will need to modify that code to be able to read in an additional file to populate the Size properties. The data file will be a simple series of names and size values, e.g.:

```
a-1 3
c-3 14
d-5 28
```

The sizes are decimal integer values. After reading the size data, you should be able to print out the Sectors (displaying at least their name and size) ordered by either Name or Size.

Two sectors will be assigned special sizes. The Sector assigned a size of 0 will be signify the 'Start', and the Sector assigned the value of 100 will signify the 'Finish'. You need to determine a path from the start to the finish. Using one of the algorithms discussed in class, or one of your own creation, follow the directional links in your Sectors and then display the path as a series of node names and directions when it is found. The direction value should be one of the 4 numeric values used in program 4 (N=1,E=2,S=4,W=8) The example below would be the output of a route from a-1 to j-5 where a-1 was the start and j-5 was the finish. Note that the direction field for j-5 is 0, which signifies that it was the exit.

```
a-1 4
d-5 8
e-1 8
x-4 1
J-5 0
```

You may choose to use your program 4 code as a base, use the provided solution as a base or any combination of the two.

In summary, your code will perform the following functions (in addition to the program 4 functionality):

- •Set the Size attribute of one or more Sectors via a second data file.
- •Display all Sectors (as at least 'name' and 'size') ordered by Size (Highest to Lowest)
- •Find a Path from the Sector with a Size of 0 to the Sector with a Size of 100 by following the directional links in the set of Sectors established by reading the data files.
- •Display that path as an ordered list of nodes and direction traveled.

Extra Credit

Extra points (out of 100) will be given in the amounts listed below for accomplishing the listed tasks, the maximum points award will be 200 in any combination.

5% - Calculate cost of trip by summing Sizes of the sectors between the start and the finish

10% - Find a path, even when a 'loop' might exist.

10% - Find multiple unique paths.

10% - Find path with the least cost.

10% - Find the path that travels through the fewest nodes.

100% - Also implement in Objective-C (using Objects for Tree, Node, and Sector, use NSArray of NSArrays (or similar) to store 'grid' of Sectors). ***Note: this means writing the C++ version and an Objective-C version***.

Primary learning objectives:

- • Using Advanced ADTs (Trees, Linked Lists, graphs)
- • Searching and Sorting ADTs

Ancillary learning objectives:

- • Implementing a complex algorithm
- • Following a 'spec'
- • Evaluating your code vs others