**Phase Two**

**Due: Friday, February 7th *immediately after class***

This assignment is the second of 4 phases to code graph theory! Now that we can represent graphs in our computers, we have some other details we need to take care of before we get into the real algorithm. In this phase, you only have to code class, the **Path** class. We will use this class to help keep track of the paths we’ve taken as we traverse graphs.

You will turn in two classes for phase two.

1. Path

2. A tester

You have been provided with no new files, but everything from phase one is still online.

Again, make sure to test as you go!

**CLASS: Path**

The Path needs to keep track of a path of vertexes in some list, the total length of the edge weights in the path (referred to as path distance from now on), and the weight of the last edge in the path. You need the following functionality:

* –  A constructor that takes in one vertex and makes a default path out of this vertex. This constructor will only be used in the beginning of our algorithm for the starting vertex.
* –  Another constructor that takes in an existing Path and copies the data over. Make sure to deep copy the path list!
* –  A third constructor that takes a Path and a vertex. It will copy the existing path and add a vertex to it. Feel free to use the previous constructor for this.
* –  Ability to add a vertex to the path. Note, make sure you update the path distance and the last edge weight.
* –  Getters for the path list, last edge weight, and path distance
* –  Getter for the number of vertexes in the path
* –  Allow the user to get a vertex in the path by index

– A getter for the last vertex in the path

– toString in the following format (refer to solution files)

Path: vertex->vertex->vertex->vertex-> Distance: path distance

The most common error here is not deep copying the path of the previous path. Remember, deep copying means making a new list and copying all the elements over (again, look into the API to make life easier). Obviously, don't deep copy the vertexes themselves.

To make sure you deep copied correctly, make a Path with the first constructor, then make another with the third constructor. Print both of them out. Are they the same? If so, you did not deep copy.

If you don't deep copy, you are gonna have a bad time!

**CLASS: Tester/Runner/Whatever**

Test test test!