Lab 6: Begin graph algorithms in Java! Practice for last homework, posting by Wednesday's class.

Read: Graphs chapter, pp. 685-713.

For this lab, I want you to get familiar with Carrano's Graph class in Java. His Graph class requires you to define the size of the graph (the number of vertices) right away, but lets you add and remove edges between them. His edges are weighted (the weights are integers). Here's how the Graph keeps track of this information. It remembers the number of vertices and the number of edges it's currently holding:

```
private int numVertices;   // number of vertices in the graph
private int numEdges;      // number of edges in the graph
```

It then keeps a private variable called adjList (short for adjacency list). This uses a built-in Java class called an ArrayList, which works like an array that can resize on demand. ArrayList is a template class (note the <>). The second template class in here is a Map class, called a TreeMap. You haven't seen Maps before, but they store a pair of Items; the first Item is the key, and the second Item is the value. Both key and value are integers, here.

```
private ArrayList<TreeMap<Integer, Integer>> adjList;
```

Suppose we have a 5-vertex graph with one edge between vertex 0 and vertex 2, and suppose this edge has a weight of 4. Its adjList would look like this:

ArrayList vertices	Map contains pair	
[0]	[2, 4]	// vertex 0 has edge to vertex 2, weight 4
[1]		
[2]	[0, 4]	// vertex 2 has the same edge to vertex 0, weight still 4
[3]		
[4]		

Notice that this edge appears twice (the graph edges are undirected).

Your demo code in GraphTester.java builds edges in the shape of a 5-point star. Play around with making Graphs of 9-point stars, 15-point stars, and 131-point stars. You will probably want to use a loop to add the edges. (You don't need to turn this in, but do need to understand how it works.)

If you started at any point of the star, like vertex 0, how would you trace out the full path 0-2-4-1-3 defined by the edges? How would you avoid going in circles? Your last homework will tackle these problems. Your TA can help you trace this out for a few simple examples.