Homework Turnin

Account:	6G_06 (rgalanos@fcps.edu)
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	12-06
Receipt ID:	0d9b57161c92a1a7d624b02928d9c9b5

Warning: Your turnin is 2 days late. Assignment 12-06 was due Friday, June 16, 2017, 11:59 PM.

Turnin Successful!

The following file(s) were received:

TJGraphAdjListWeighted.java (4002 bytes)

```
1. //name:   date:
2. //for use with Graphs6: Dijkstra
3. //             Graphs7: Dijkstra with Cities
4.
5. import java.io.*;
6. import java.util.*;
7.
8. class Edge {
9.    public final wVertex target;
10.   public final double weight;
11.
12.   public Edge(wVertex argTarget, double argWeight) {
13.      target = argTarget;
14.      weight = argWeight;
15.   }
16. }
17.
18. class wVertex implements Comparable<wVertex>, wVertexInterface
19. {
20.   private final String name;
21.   private ArrayList<Edge> adjacencies;
22.   private double minDistance = Double.POSITIVE_INFINITY;
23.
24.    /*  enter your code here   */
25.   public wVertex(String s)
26.   {
27.      name = s;
28.      adjacencies = new ArrayList<Edge>();
29.   }
30.   public String toString()     //just return the name
31.   {
32.      return name;
33.   }
34.   public String getName()
35.   {
36.      return name;
37.   }
38.   public double getMinDistance()
39.   {
40.      return minDistance;
41.   }
42.   public void setMinDistance(double m)
```

```
43.    {
44.       minDistance = m;
45.    }
46.    public ArrayList<Edge> getAdjacencies()
47.    {
48.       return adjacencies;
49.    }
50.    public int compareTo(wVertex other)
51.    {
52.       if(minDistance<other.getMinDistance())
53.          return -1;
54.       else if(minDistance>other.getMinDistance())
55.          return 1;
56.       else
57.          return 0;
58.    }
59.    public void addEdge(Edge e)
60.    {
61.       adjacencies.add(e);
62.    }
63. }
64.
65. interface wVertexInterface
66. {
67.    public String toString();
68.
69.    public String getName();
70.
71.    public double getMinDistance();
72.
73.    public void setMinDistance(double m);
74.
75.    // public wVertex getPrevious();         //Graphs 7
76.
77.    // public void setPrevious(wVertex v);   //Graphs 7
78.
79.    public ArrayList<Edge> getAdjacencies();
80.
81.    public int compareTo(wVertex other);
82. }
83.
84.
85. public class TJGraphAdjListWeighted implements TJGraphAdjListWeightedInterface
86. {
87.    private ArrayList<wVertex> vertices = new ArrayList<wVertex>();
88.    private Map<String, Integer> nameToIndex = new HashMap<String, Integer>();
89.
90.    /*  enter your code here   */
91.
92.    public List<wVertex> getVertices()
93.    {
94.       return vertices;
95.    }
96.
97.    public wVertex getVertex(int i)
98.    {
99.       return vertices.get(i);
100.   }
101.
102.   public wVertex getVertex(String vertexName)
103.   {
104.      return getVertex(nameToIndex.get(vertexName));
105.   }
106.
107.   public void addVertex(String v)
108.   {
109.      vertices.add(new wVertex(v));
110.      nameToIndex.put(v, vertices.size()-1);
111.   }
112.
113.   public void addEdge(String source, String target, double weight)
114.   {
115.      vertices.get(nameToIndex.get(source)).addEdge(new Edge(new wVertex(target), weight));
116.   }
117.
118.   public void minimumWeightPath(String vertexName)
119.   {
120.      PriorityQueue<wVertex> pq = new PriorityQueue<wVertex>();
121.      wVertex source = getVertex(vertexName);
122.      source.setMinDistance(0);
123.      pq.add(source);
```

```
124.
125.        while(!pq.isEmpty())
126.        {
127.           wVertex vertex = pq.remove();
128.           ArrayList<Edge> neighbors = vertex.getAdjacencies();
129.           for(Edge e: neighbors)
130.           {
131.              if(vertex.getMinDistance()+e.weight<getVertex(e.target.getName()).getMinDistance())
132.              {
133.                 wVertex v =getVertex(e.target.getName());
134.                 v.setMinDistance(vertex.getMinDistance()+e.weight);
135.                 pq.add(v);
136.              }
137.           }
138.        }
139.     }
140.
141.
142. }
143. interface TJGraphAdjListWeightedInterface
144. {
145.    public List<wVertex> getVertices();
146.
147.    public wVertex getVertex(int i);
148.
149.    public wVertex getVertex(String vertexName);
150.
151.    public void addVertex(String v);
152.
153.    public void addEdge(String source, String target, double weight);
154.
155.    public void minimumWeightPath(String vertexName);   //Dijkstra's
156.
157.    /*  Graphs 7 */
158.
159. //  public List<wVertex> getShortestPathTo(wVertex v);
160.
161. //  public TJGraphAdjListWeighted graphFromEdgeListData(File vertexNames, File edgeListData) throws FileNotFound
162.
163. }
```