Homework Turnin

Account:	6G_06 (rgalanos@fcps.edu)
Section:	6G
Course:	TJHSST APCS 2016-17
Assignment:	12-00
Receipt ID:	34dabeff8e557da6e3c908d9a8f3ddd4

Turnin Successful!

The following file(s) were received:

TJGraphAdjMat.java (2528 bytes)

```java
//name:   date:
// resource classes and interfaces
// for use with Graphs0: Intro
//                Graphs1: Wardhall
//                Graphs2: Floyd
import java.util.*;
import java.io.*;

interface AdjacencyMatrix
{
   public void addEdge(int source, int target);
   public void removeEdge(int source, int target);
   public boolean isEdge(int from, int to);
   public void displayGrid();
   public int edgeCount();
   public List<Integer> getNeighbors(int source);

}

interface Warshall
{
   //User-friendly functionality
   public boolean isEdge(String from, String to);
   public Map<String, Integer> getVertices();
   public void readNames(String fileName) throws FileNotFoundException;
   public void readGrid(String fileName) throws FileNotFoundException;
   public void displayVertices();
   //Actual Warshall Algorithm
   public void allPairsReachability();
}

interface Floyd
{
   public int getCost(int from, int to);
   public int getCost(String from, String to);
   public void allPairsWeighted();
}

public class TJGraphAdjMat implements AdjacencyMatrix //,Warshall,Floyd
{
   private int[][] grid = null;   //adjacency matrix representation
   private Map<String, Integer> vertices = null;

   /*  enter your code here  */
   public TJGraphAdjMat(int size)
   {
      grid = new int[size][size];
      vertices = new HashMap<String, Integer>();
   }

   public void addEdge(int source, int target)
   {
      grid[source][target]=1;
   }

   public void removeEdge(int source, int target)
   {
      if(grid[source][target]==1)
         grid[source][target]=0;
      else
         System.out.println("Thats not an edge");
   }

   public boolean isEdge(int from, int to)
   {
      return grid[from][to]==1;
   }

   public void displayGrid()
   {
      for(int[] x: grid)
      {
         for(int y: x)
         {
            System.out.print(y+" ");
         }
         System.out.println("");
      }
   }

   public int edgeCount()
   {
      int count = 0;

      for(int[] x: grid)
      {
         for(int y: x)
         {
            if(y==1)
               count++;
         }
      }
      return count;
   }
   public List<Integer> getNeighbors(int source)
   {
      List<Integer> list = new ArrayList<Integer>();
      for(int i=0;i<grid[source].length;i++)
      {
         if(isEdge(source, i))
            list.add(i);
      }
      return list;
   }
}

```