===Lab Info===

* 100 points

* Due 11:59pm on Thursday 12/10/2015 for Monday Lab and Wednesday Lab.

==Assignment==

In this assignment you will work on designing a class for finding the Minimum Spanning Tree (MST) in a graph. One of the applications of minimum spanning tree is in the communication network. Consider that there are several installations; there is a cost if we want to connect two installations with each other, and we want to connect the installations in a way that all installations are connected together and the cost becomes minimum. MST is mentioned in the US legal code, and AT&T Company is responsible for connections in the minimum cost manner, which is a MST.

For implementing Minimum Spanning Tree, Kruskal and Prim algorithm should be used. In Kruskal algorithm the edges are chosen from the shortest and add to the structure. If by choosing an edge, a cycle occurs that edge will not be chosen. In order to find the cycles, disjoint sets must be used. Also for choosing the shortest edge you need to sort the edges, any priority queue or sorting algorithm can be used.

For Prim algorithm, we should start from an arbitrary vertex. **Consider vertex 0 as the first vertex.** The shortest edge that is connected to vertex 0 is chosen and is added to the chosen edges. The other vertex is also added to the list of chosen vertices. In prim algorithm there is no need to check for cycle. When an edge has the shortest length it is chosen only if just one its vertices belong to the chosen vertices, otherwise choosing that vertex cause a cycle. In this implementation a priority queue is also needed to keep track of the shortest edge each time. Another method for implementing Prim algorithm is to use a table and labeling technique, in this case there is no need to use a priority queue.

The adjacency matrix should be read from the file data.txt. Then your program should output the edges which have been used to create the Minimum Spanning Tree. The first number in the file is the number of test cases, the next number shows the number of the nodes in the first graph and then you will have the adjacency matrix corresponding to the graph. **A 0 in the adjacency matrix shows that there are no connections between the two nodes.**

Also answer to these questions:

1) Should the tree after applying the two methods always be the same? (describe)
2) Which of these two algorithms is faster?

3) Based on your initial experiment on data set #1, do you think that the resulting, i.e., communication network is optimal? Will AT&T satisfy the U.S. legal code with this communication network?
4) Now, run your code on data set #2 which contains some additional nodes. What is the cost of this communication network? Will AT&T satisfy the U.S. legal code with this communication network?
5) Was your hypothesis in Question 1 correct? Why or why not?

6) Why do you think AT&T would consider adding an additional node into their communication network (such as in Question 4)? What are the pros and cons of doing so?

```
===Output===

data1.txt elements :

2

3

0   5    7

5    0    6

7    6    0

5

0      5    7   11   8

5      0    6     0   12

7    6    0     9     0

11    0    9     0    7

8     12   0     7    0

..........................................................

data2.txt elements :

2

4

0   5    7    4

5    0    6   3

7    6    0   3

4    3    3   0

6

0      5    7   11   8    0

5      0    6     0   12    4

7    6    0     9    0     3
```

```
11    0    9     0    7      5
8     12   0     7    0      0
0      4   3     5    0      0
```

....

Output for data1.txt:

```
Graph1:
Kruskal:  (0, 1) (1, 2)    Cost = 11
Prim:  (0, 1) (1, 2)   Cost = 11
Graph2:
Kruskal: (0, 1) (1,2) (3,4) (0, 4)     Cost = 26
Prim: (0, 1) (1,2) (0,4) (3,4)    Cost = 26
```

Output for data2.txt:

```
Graph1:
Kruskal:  (1,3) (2,3) (0,3)    Cost = 10
Prim:  (0,3) (1,3) (2,3)   Cost = 10
Graph2:
Kruskal: (2,5) (1,5) (3,5) (0, 1) (3,4)    Cost = 24
Prim: (0, 1) (1, 5) (2,5) (3,5) (3,4)       Cost =24
```

The orders of the edges are important

===Files===

* Files to include in folder:

** All source files

** A functioning makefile

** data.txt

* A pdf file for answering the questions

* Folder name: Lastname_Lab11

* Compressed file name: Lastname_Lab11.zip (or .rar or .tar.gz)

* Executable name: lab11