Assignment 4: Minimum Spanning Tree

Implement the Prim's algorithm for finding the Minimum Spanning Tree (MST) of a given graph. The graph will be connected.

Direction

Create a class named `PrimMST` with a constructor which will take the *adjacency matrix* representation of the graph and an integer indicating *start node*.

Create a method named `getMST` inside the class. This method will return a 2-d array and will not take any parameter. The array will be the *adjacency matrix* representation of the TREE just calculated by the algorithm.

The nodes will be represented using integers ranging from `0` to `n-1` for a graph containing `n` nodes.

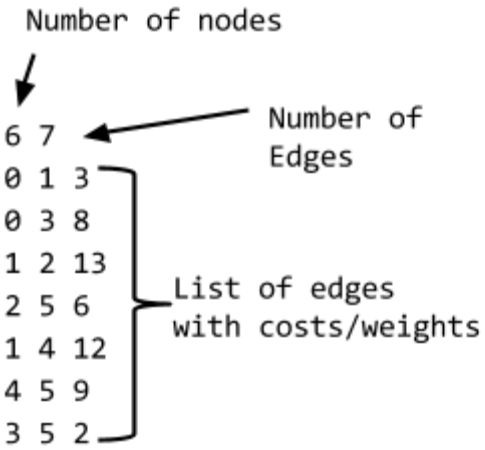
As earlier, you can add any method you feel needed to those classes. Number of nodes in the graphs will be no more than 1000. Calculate the maximum number of edges possible in a graph with that many nodes. That is the limit for number of edges. Cost/weight for edges will be distinct and range from 1 to 1000000. So, MST will be unique for a particular data set.

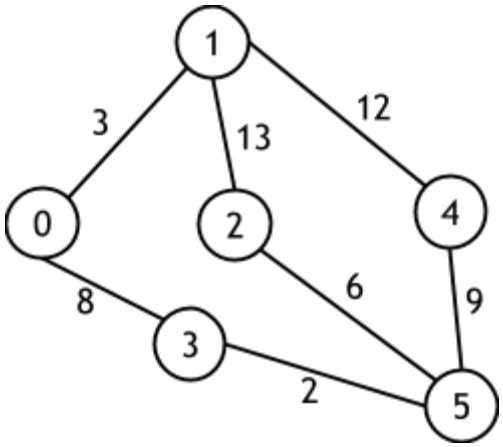
Way to solutions

Follow the class lectures and resources listed in the Reading Materials section at the end of the document. Follow the reading materials section carefully. Use them to understand the concepts. They provided slightly advanced version of the algorithm. That is not required for completing this assignment.

Sample Input/Output

Some set of input and output are given here. Format is explained with the example below.



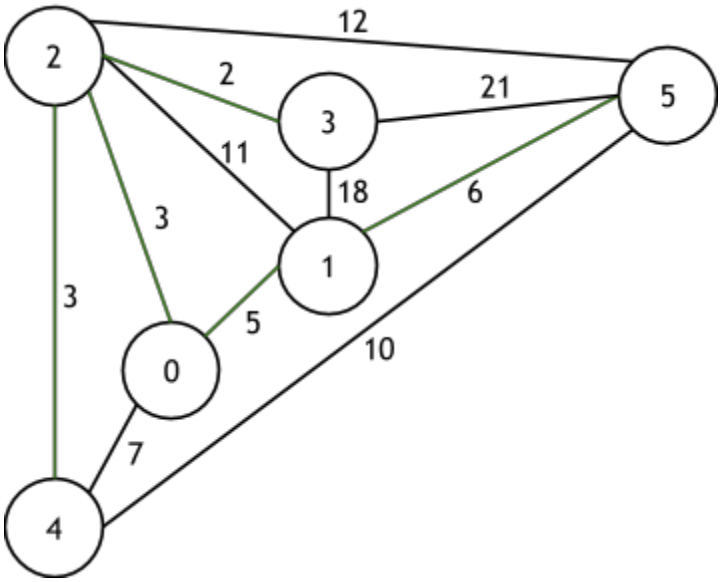


The first line of the input will contain 2 integers. They are number of nodes and number of edges respectively. Following that there will be as many lines as there are edges. Each line will contain the two associated nodes of the corresponding edge followed by the weight assigned to it.

Sample Input:

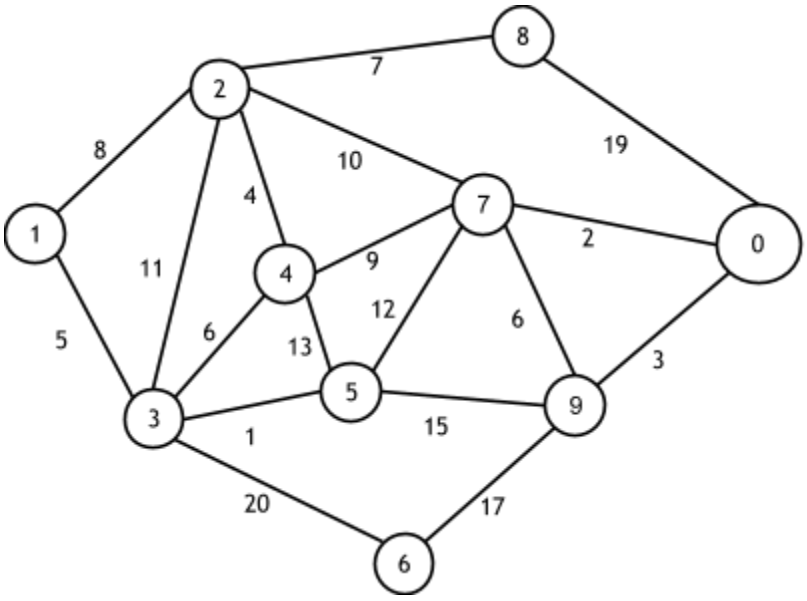
Set 1:

```
6 11
0 1 5
0 2 3
0 4 7
1 2 11
1 5 6
1 3 18
2 3 2
2 5 12
2 4 3
3 5 21
4 5 10
```



Set 2:

```
10 18
1 3 5
1 2 8
2 3 11
2 4 4
3 4 6
4 5 13
3 5 1
4 7 9
5 7 12
2 7 10
2 8 7
5 9 15
3 6 20
6 9 17
7 9 6
7 0 2
8 0 19
9 0 3
```



Sample Output:

Set 1:

```
0 5 3 0 0 0
5 0 0 0 0 6
3 0 0 2 3 0
0 0 2 0 0 0
0 0 3 0 0 0
0 6 0 0 0 0
```

Set 2:

```
0 0 0 0 0 0 0 2 0 3
0 0 0 5 0 0 0 0 0 0
0 0 0 0 4 0 0 0 7 0
0 5 0 0 6 1 0 0 0 0
0 0 4 6 0 0 0 9 0 0
0 0 0 1 0 0 0 0 0 0
0 0 0 0 0 0 0 0 0 17
2 0 0 0 9 0 0 0 0 0
0 0 7 0 0 0 0 0 0 0
```

```
3 0 0 0 0 0 17 0 0 0
```

Evaluation

Submission will be evaluated using following criteria (with distribution of marks)

Criteria	**Marks**
Created necessary java files (according to directions)	10
Implementation of Prim's algorithm	70
Creating adjacency matrix of the TREE	20
Total	**100**

For Incorrect Implementations:

Implementation of Prim's algorithm *	5-55
Creating adjacency matrix of the TREE *	0-15

* Marking depends on evaluator.

***Notes on copying other's code*: If anybody found copying solution code from other student(s), all of them will be penalized. Penalty includes**

- ➢ ***Assigning ZERO as mark for the solution submitted***
- ➢ ***Assigning ZERO as mark for the best submission among all other submissions. (assigned at the end of semester)***

Copying solutions from the Internet will also incur similar penalties.

Submission & Contact

Submission Type	Individual
Submission Deadline (Final): *	June 15, 2014

* NO extension on deadline will be allowed.

Students are encouraged to use Google Group for communicating to resolve their issues (instead of personal mails.)

Group Address: https://groups.google.com/forum/#!forum/algorithmfiesta2014

Reading Materials

- *Computer Algorithms*, By Ellis Horowitz , Sartaj Sahni, Sanguthevar Rajasekaran. ISBN: 9780929306414 **Page-216**
- *Introduction to Algorithms*, By Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest and Clifford Stein. ISBN: 9780262033848 **Page-561**