# Programming Assignment 3

CMPE 250, Data Structures and Algorithms, Fall 2011

Instructor: A. T. Cemgil
TA's: Barış Kurt, Murat Arar, Zeynep Gözen Sarıbatur

Due: 14 Dec 2011, 15:00 Sharp

## Problem Definition

In this project you are going to implement the *Prim*'s and *Kruskal*'s minimum spanning tree (MST) algorithms in *C language*.

## Input and Output Format

Your program will read the graph from a text file, and write the result in another text file. The selection of the MST algorithm will be done by *-p* and *-k* flags. The command for running your program will be like this:

```
./your_program input.txt -p prim_output.txt
./your_program input.txt -k kruskal_output.txt
```

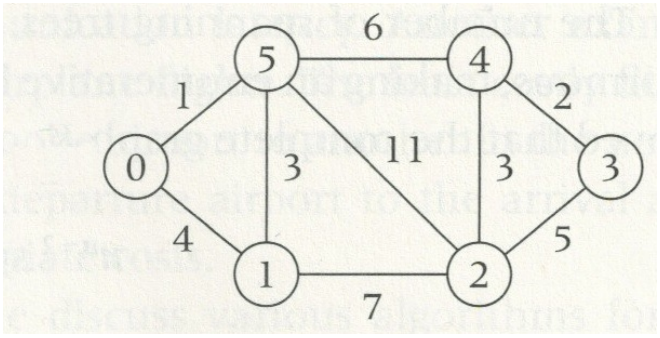
In the input file, the first line will be the number of nodes, and remaining lines are the symmetric adjacency matrix representing the undirected graph. Following is the representation for the graph in Figure 1(a).

```
6
0  4  0  0  0  1
4  0  7  0  0  3
0  7  0  5  3  11
0  0  5  0  2  0
0  0  3  2  0  6
1  2  11 0  6  0
```

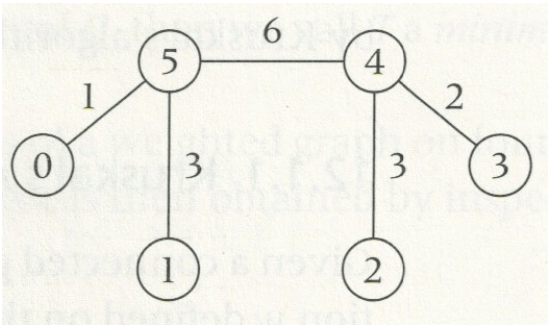
In the output file, the first line will be the number of nodes, second line will be the total weight of the edges in the tree, and remaining lines are the symmetric adjacency matrix representing the *unweighted* undirected graph which is actually the minimum spanning tree. Following is the representation for the MST of the example input file which is also shown in Figure 1(b).

```
6
15
0 0 0 0 0 1
0 0 0 0 0 1
0 0 0 0 1 0
```

```
0 0 0 0 1 0
0 0 1 1 0 1
1 1 0 0 1 0
```




(a) A sample input Graph




(b) The MST of the input graph.

## Submission & Grading

- **What, How and When to submit**
  - You should submit your project in electronic form.
  - You should compress your source code (.cpp and .h files) in a zip file, name it as [pr]_[#]_[Student ID] (e.g. `pr_3_200700803.zip`), and email to `bucmpe250@gmail.com`.
  - The subject of your e-mail must also be the same as your zip file (e.g. `pr_3_200700803`)
  - Your zip file should NOT contain any executable file (.exe) or any folder. If you sent multiple e-mails, only the last one will be taken into account.
  - The deadline is 14 Dec 2011, 15:00 Sharp. Emails tagged later won't be considered.
- **Grading**
  - Warning: All source codes are checked automatically for similarity with other submissions and exercises from previous years. Make sure you write and submit your own code.
  - Your program will be graded based the correctness of your output and the clarity of the source code. Correctness of your output will be tested automatically so make sure you stick with the format described above.
  - There are several issues that makes a code piece 'quality'. In our case, you are expected to use C++ as powerful, steady and flexible as possible. Use mechanisms that affects these issues positively.
  - Make sure you document your code with necessary inline comments, and use meaningful variable names. Do not over-comment, or make your variable names unnecessarily long.
  - Try to write as efficient (both in terms of space and time) as possible. Informally speaking, try to make sure that your program completes under 20 seconds.