



## CNG315 Homework 2 (Deadline: 10/12/2015)

In this homework, you are expected to implement Prim's Minimum Spanning Tree algorithm.

### Specifications:

Implement Prim's MST algorithm as follows:

- Use adjacency matrix to represent the graph.
- Use unsorted array to store the "key" array values (i.e., do not use heap or any other data structure to perform extract minimum valued key at each iteration, but just use  $O(N)$  algorithm and unsorted key array to extract minimum valued key, and perform decrease key operations in  $O(1)$  time on that array).
- The complexity of your algorithm will be  $O(N^2)$ .

The input and the output will be as follows:

### INPUT

$N$   $M$  -- number of vertices and number of edges  
 $v_i$   $v_j$   $w(v_i, v_j)$  -- vertex  $i$ , vertex  $j$  and the weight of the edge between vertex  $i$  and vertex  $j$   
... -- all  $M$  edges

(Note that, vertices will be numbered from 1 to  $N$ .)

### OUTPUT

$W$  -- total weight of the MST  
 $v_i$   $v_j$   $w(v_i, v_j)$  -- vertex  $i$ , vertex  $j$  and the weight of the edge between vertex  $i$  and vertex  $j$   
.. -- all  $N$  edges in the MST

### Dimensions

$N \leq 100$ ,  $M \leq 10000$   
 $0 < w(v_i, v_j) < 100$

### Sample Input (for lec16.pdf on odtuclass, pg. 27-39)

```
8 10
1 2 7
1 4 15
1 8 10
2 3 9
2 5 12
2 6 5
5 6 6
6 7 14
6 8 8
7 8 3
```

### Sample Output (for lec16.pdf on odtuclass, pg. 27-39)

```
53
1 2 7
1 4 15
2 3 9
2 6 5
5 6 6
6 8 8
7 8 3
```

### Additional Information

- **Programming Language:** You must code your program in C/C++. Your submission will be compiled on my computer using Dev C++.
- **Cheating:** In case of cheating, the university regulations will be applied.
- **Modularity:** Please follow the modular programming approach. In C++ we use functions referred to modules to perform specific tasks that are determined/guided by the solution. Remember the following tips and please don't forget that I will grade the modularity of your program!
  - Modules can be written and tested separately!
  - Modules can be reused!
  - Large projects can be developed in parallel by using modules!
  - Modules can reduce the length of the program!
  - Modules can also make your code more readable!

If you have any questions about the homework, please send me an e-mail.

Have fun,  
Baris Engin Sonmez  
Course TA