

Programming Assignment #1

Due: 2/24/2012

Total 100 points.

1. Implementation of Prim's algorithm for Minimum Spanning Tree. [50 points]

Implement the Prim's algorithm with array data structure as described in slide 12 of the file 04mst.ppt. Your program should have a runtime complexity of $O(n^2)$ and should be as efficient as possible. It should be able to handle up to 6K nodes and 18M edges. Please use adjacency list to represent the graph. You may use either C or C++ for the implementation.

The input weighted undirected graph is specified in a text file in the format below:

```
<# of nodes>
<# of edges>
<node 1 of edge 1> <node 2 of edge 1> <weight of edge 1>
<node 1 of edge 2> <node 2 of edge 2> <weight of edge 2>
...
<node 1 of edge m> <node 2 of edge m> <weight of edge m>
```

For example, the following file describes a graph with 3 nodes (indexed 1, 2 and 3) and 2 edges ({1,2} with weight 0.5 and {2,3} with weight 2.2).

```
3
2
1 2 0.5
2 3 2.2
```

For the output, your program should display the total weight of the MST and write a file representing the MST in the same format as the input file above.

Several graphs will be posted to test your program. Please make a table to report, for each graph posted, the number of nodes in the graph, the number of edges in the graph, the runtime, and the total weight of the MST. Please also report the processor model and clock frequency of the machine that you use to obtain the runtime. Please email a zip file named prog01-prim.zip containing your source code and all output files to the TA.

2. Implementations of Kruskal's algorithm for Minimum Spanning Tree. [50 points]

Repeat Q.1 except:

- You are implementing Kruskal's algorithm here. Please implement the array-based Union-Find data structure as described in slides 15-16 of the file 04mst.ppt and Claim (4.23) in page 153 of the textbook. The resulting program should have a time complexity of $O(m \log n)$.
- The file you email to the TA should be named prog01-kruskal.zip.