Lab Assignment 3

Implementing Minimum Spanning Tree (MST) Algorithm

(Due: Thursday May 7, 2020; 11:59pm)

1. Overview

We have discussed the usage of minimum spanning tree when you try to extend LAN segments with link-layer switches (bridges). In this programming assignment, you will be implementing Minimum Spanning Tree (MST) Algorithm with either C or C++.

1.1 Input File

- Test input file will be randomly generated when your program will be tested.
- Test input file format: represent current status of entire network
  - Test input file contains link information and cost of between link
    - C[i][j] = *n* if there is direct link from node i to node j,
      where 0 <= *n* < 100
      = 1073741824 (=2^30) otherwise
  - The first line of test input file has the total number of node in the network
  - Following each line contains link cost between i and j.
- Sample test input file:

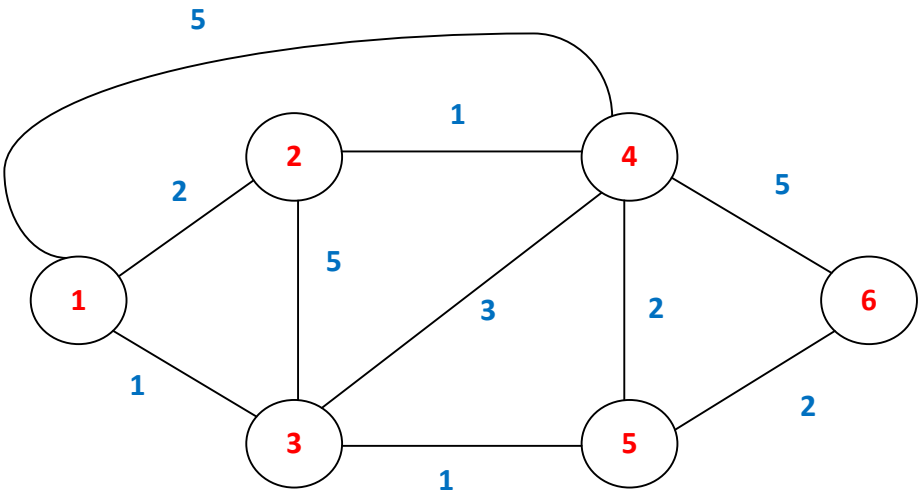

Figure 1: Example Network Topology

Sample test the input file of Figure 1 will be as follows:

```
6
1 1 0
1 2 2
1 3 1
```

```
1 4 5
1 5 1073741824
1 6 1073741824
2 1 2
2 2 0
2 3 5
2 4 1
2 5 1073741824
2 6 1073741824
3 1 1
3 2 5
3 3 0
3 4 3
3 5 1
3 6 1073741824
4 1 5
4 2 1
4 3 3
4 4 0
4 5 2
4 6 5
5 1 1073741824
5 2 1073741824
5 3 1
5 4 2
5 5 0
5 6 2
6 1 1073741824
6 2 1073741824
6 3 1073741824
6 4 5
6 5 2
6 6 0
```

1.2. Your program input arguments

Your program should be able to take two inputs as follows:

Unixprompt> mymst <test-input-file>

Where: <test-input-file> : test input file name

2. Output

1) Minimum Spanning Tree (T) and Minimum Cost
2) Display total execution time in *ms* (millisecond) from reading test input files to find a MST (Minimum Spanning Tree)

Sample Output based on Figure 1

```
Total Execution time = 10 ms
Minimum Cost = 7
Minimum Spanning Tree (T) : (1,2) → (2,4), (1,3) → (3,5) → (5,6)
```

2.3 Program testing

1) Your program should run on csegrid without any problems
2) You can assume that the maximum number of node is 100

3. Grading

The maximum possible point for the assignment is 30. This programming assignment will be graded by following criteria.

- Completeness: 20 points
  - If your program works correctly, you will get 20 points
- Completeness of submission: 5 points
- Efficiency of program: 5 points max
  - Top 10% fastest programs get 5 points
  - Top 20% fastest programs get 4 points
  - Others get 3 points

4. How to submit

Please do the followings when you submit your programming assignment:

- Create a tar or zip file that contains your written source code, makefile and readme. DO NOT INCLUDE EXECUTABLES AND OBJECT FILES.
- Submit through Canvas submission link