COP 3530, Fall 2015

Data Structures and Algorithms

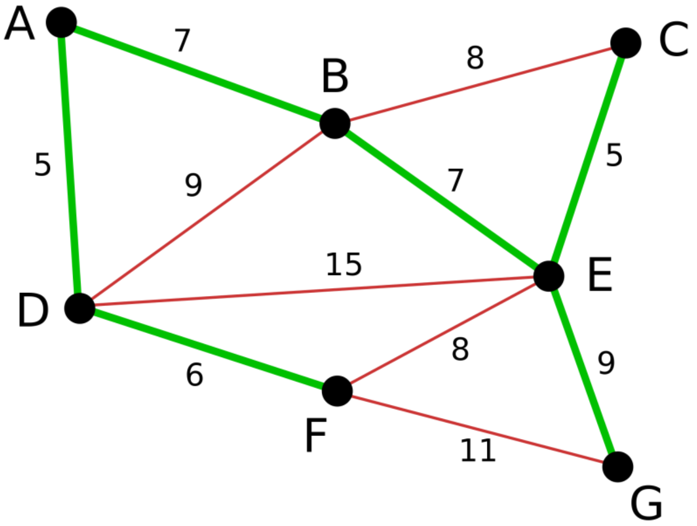
Assignment 4: Minimum Spanning Trees

Due: Monday, Dec. 14, 11:55 p.m. via Canvas

Description:

For a weighted, connected, undirected graph, a Spanning Tree is a sub-graph that includes all the vertices of the original graph, and is also a tree. A Minimum-Cost Spanning Tree is a spanning tree with the smallest sum of its edges' costs. Uses of Minimum-Cost Spanning Trees range from Taxonomy to Clustering analysis.

In this assignment, you will implement Kruskal's, Boruvka's (Sollin's) , and Prim's algorithms to produce Minimum-Cost Spanning Trees from an undirected graph. You may implement Kruskal's algorithm without the Union-Find operation.



Sample input and output (User Input is represented in RED font):

```
Enter number of Node and Edges(s):
4 5
Enter Node A and Node B and Undirected Edge Weight(s):
0 1 5
1 3 4
3 2 2
2 0 7
1 2 10
Enter the start Node:
0
```

```
Prim's MST:
(0, 1)
(1, 3)
(2, 3)
Total Weight:
11

Kruskal's MST:
(2, 3)
(1, 3)
(0, 1)
Total Weight:
11

Boruvka’s MST:
(0, 1)
(2, 3)
(1, 3)
Total Weight:
11
```

Note: Edge(s) are undirected and weighted.

We will test your submissions by following these steps (commands on thunder.cise.ufl.edu):

```
1. tar xvf “<LastName_FirstName>_UFID.tar”

2. make

3. ./MST < our_input
```

Deliverables:

- Your submission tarball(.tar archive file) should be named <LastName_FirstName>_UFID.tar and must contain the following files: MST.cpp, Makefile, and <LastName_FirstName>_UFID_report.pdf

- A PDF document containing your own test cases, test results, and any special diagnostics you utilized. This file shall be named <LastName_FirstName>_UFID_Report.pdf

PLEASE NOTE that **ALL** submissions **MUST** compile on **thunder.cise.ufl.edu** by using your Makefile. It is highly recommended that you upload your source code to your CISE account and test it on the thunder. Secure Remote Access to CISE machines is available. Please visit this link for more information, https://www.cise.ufl.edu/help/access/remote. We will not debug your source code or makefile. Please refer to and make note of the submission rules and policies before submitting, http://cise.ufl.edu/class/cop3530fa15/SubmissionRules.htm. After submitting on Canvas, you should verify that your submission was successful by downloading it from Canvas (to a separate location), and successfully un-tarring, compiling, and running it. YOU ARE NOT DONE UNTIL YOU DO THIS.