CSCE 221 Cover Page
Programming Assignment #6
Due December 6 by 11:59 pm to CSNet

First Name	Kevin	Last Name	Chou	UIN	224000120
User Name	k11	E-mail address	k11@tamu.edu		

Please list all sources in the table below including web pages which you used to solve or implement the current homework. If you fail to cite sources you can get a lower number of points or even zero. According to the University Regulations, Section 42, scholastic dishonesty are including: acquiring answers from any unauthorized source, working with another person when not specifically permitted, observing the work of other students during any exam, providing answers when not specifically authorized to do so, informing any person of the contents of an exam prior to the exam, and failing to credit sources used. Disciplinary actions range from grade penalties to expulsion read more: Aggie Honor System Office

Type of sources			
People			
Web pages (provide URL)			
Printed material			
Other Sources			

I certify that I have listed all the sources that I used to develop the solutions/codes to the submitted work.

"On my honor as an Aggie, I have neither given nor received any unauthorized help on this academic work."

Your Name Kevin Chou Date 12/6/15

Programming Assignment 6 Report

December 6, 2015

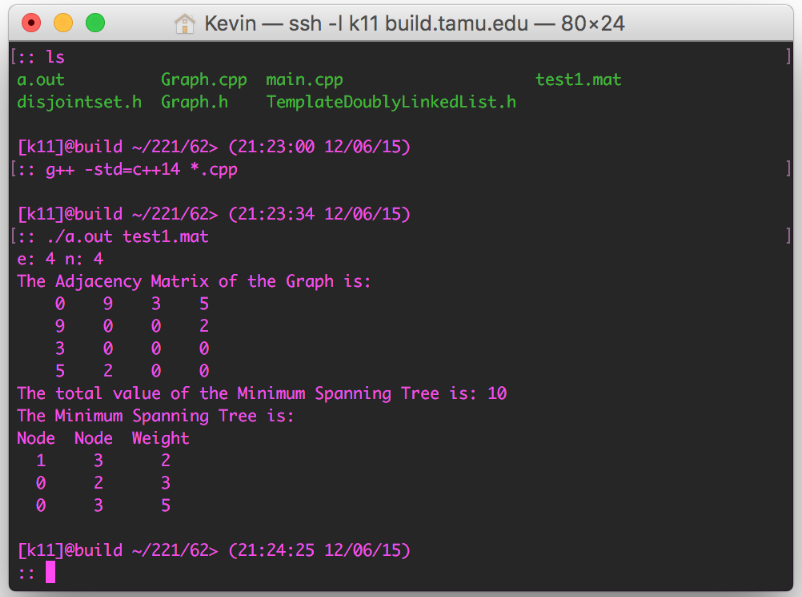
Program description and Purpose of Assignment

The purpose of the assignment is to implement a graph structure and using Kruskal's algorithm to build a minimum spanning tree. The assingment is split into two parts. Part 1 is building the graph and inserting its edges and getting the weight of the edges. Part 2 of the assignment is building the minimum spanning tree using Kruskal's algorithm.

Runtime analysis of functions

Part 1 of the program has 3 crucial functions: buildGraph(), insertEdge(), and getWeight(). buildGraph() iterates to puts n vertices into an adjacency list. This makes buildGraph() have a runtime of O(n). insertEdge() creates new edges and vertices inserts them in the corresponding index in the adjacency list. insertEdge() has a constant runtime O(1). getWeight() traverses the adjacency list and gets the weight of the edges. It goes through n edges so getWeight() has a runtime of O(n). Part 2 of the program uses 2 functions: sortEdge() and MSTlgo(). sortEdge() uses std::sort() and has a runtime of O(1). MSTlgo() is the implementation of Kruskal's algorithm. It has a runtime of O(n).

Results



Kevin — ssh -l k11 build.tamu.edu — 80×24

```
[:: ls                                                                         ]
a.out          Graph.cpp  main.cpp                     test1.mat
disjointset.h  Graph.h    TemplateDoublyLinkedList.h

[k11]@build ~/221/62> (21:23:00 12/06/15)
[:: g++ -std=c++14 *.cpp                                                       ]

[k11]@build ~/221/62> (21:23:34 12/06/15)
[:: ./a.out test1.mat                                                          ]
e: 4 n: 4
The Adjacency Matrix of the Graph is:
    0    9    3    5
    9    0    0    2
    3    0    0    0
    5    2    0    0
The total value of the Minimum Spanning Tree is: 10
The Minimum Spanning Tree is:
Node  Node  Weight
  1     3      2
  0     2      3
  0     3      5

[k11]@build ~/221/62> (21:24:25 12/06/15)
::
```