Alejandro Pena Caicedo
Ufid:3712-3113
Assignment 4

Explanation:

The 3 minimum spanning tree algorithms were implemented using two main classes, a heap class and the main class, minimum spanning tree. The purpose of the heap class was to allow the creation of heaps to store the edges of the minimum spanning trees formed by the 3 algorithms. The heap classes base was the vector library and since it was a vector of objects I created a custom bubble sort function to sort the vector heap in ascending order (to receive the edges for the mst in from least weight to greatest). The mst class was the main class being used, it contained the variables and functions required to first create the adjacency matrix given through the terminal/user. Then using the adjacency matrix, prims, boruvkas, and kruskal found their least cost edges and stored them in the heap mentioned about the previous class. I then printed out the minimum spanning tree of each algorithm as well as their respective total weights.

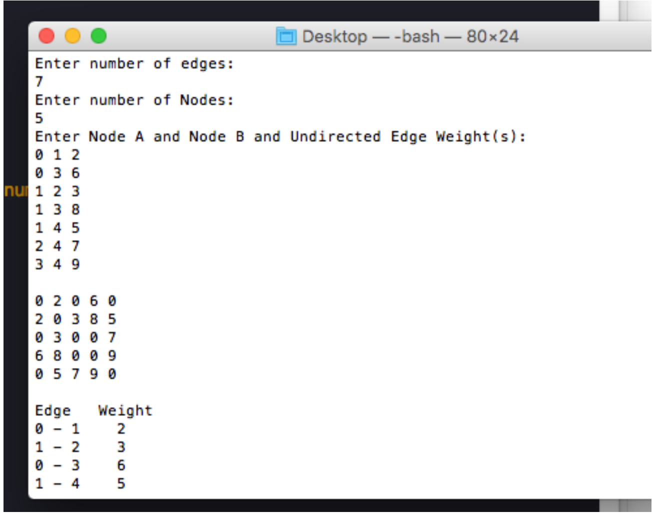
Alejandro Pena Caicedo
Ufid:3712-3113
Assignment 4

Testing Results:

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

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

Example of how Boruvka’s, Prim’s, and Kruskal’s were tested in the terminal:
Note* This is Prim’s being tested in the creation process.



```
Enter number of edges:
7
Enter number of Nodes:
5
Enter Node A and Node B and Undirected Edge Weight(s):
0 1 2
0 3 6
1 2 3
1 3 8
1 4 5
2 4 7
3 4 9

0 2 0 6 0
2 0 3 8 5
0 3 0 0 7
6 8 0 0 9
0 5 7 9 0

Edge   Weight
0 - 1    2
1 - 2    3
0 - 3    6
1 - 4    5
```

Alejandro Pena Caicedo
Ufid:3712-3113
Assignment 4