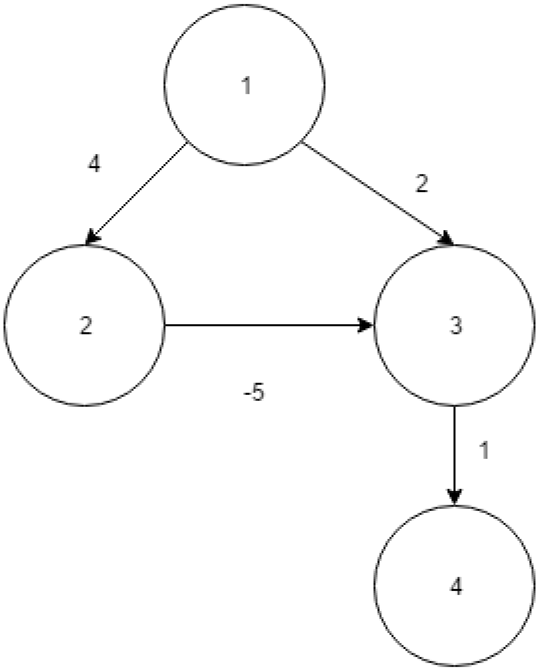
Steven Kravitsky

CS 260

Homework 8

1)



Dijkstra's algoritm can start on Node 1, poping it off the priority queue. There are two places it can go which are 1 -> 2 and 1 ->3, but it will go to 3 because it has a lower cost. So it will pop 3 off the priority queue and see 3->4, so it will pop 4 off the priority queue. Now it will go back to 2 since it will go back to the lowest remaining node, popping 2 off the priority queue. The distance to Node 3 will be updated -1. The algorithm will say that the shortest path from 1 to 4 is 1 ->3->4 because Dijkstra's algorithm does not handle negative edge weights well.

2)

To insert the edge(I, j) into the adjacency list the algorithm will go to i and scan the adjacency list to see if j already exists there. If J exsists then nothing will happen, if it does not then it will be appended to the list. The algorithm will then go to j and look for I, appending it if it doesn't exist. To delete edge(I,j) it's the same as inserting but with deleting.

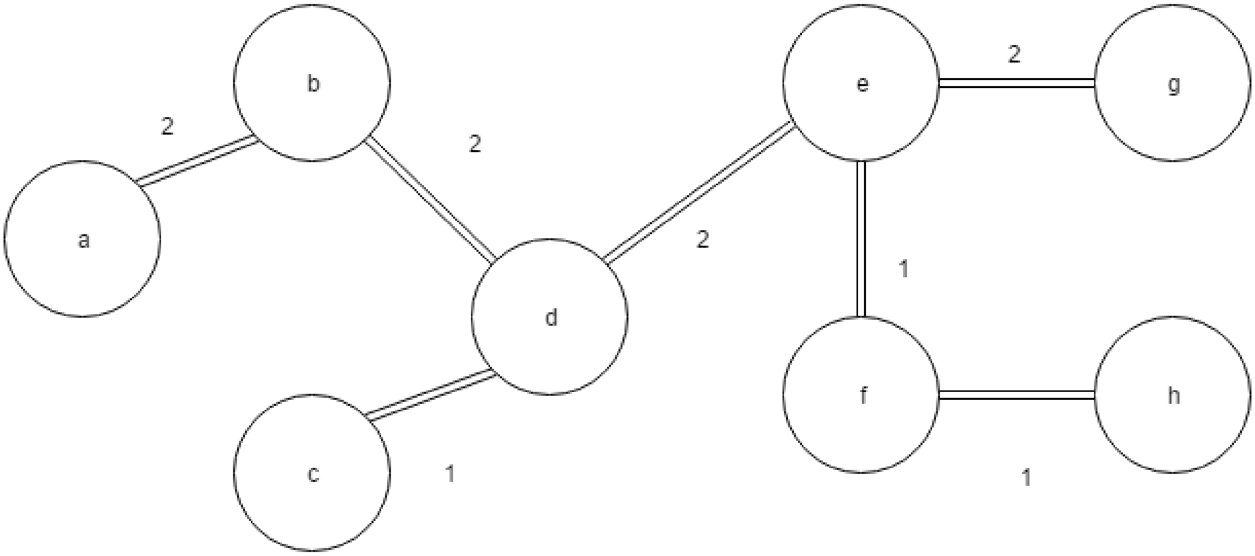
3)

```
Delete(I, J, S)
        For j in adj[i]
                Delete j.SharingEdge
                Delete j
```

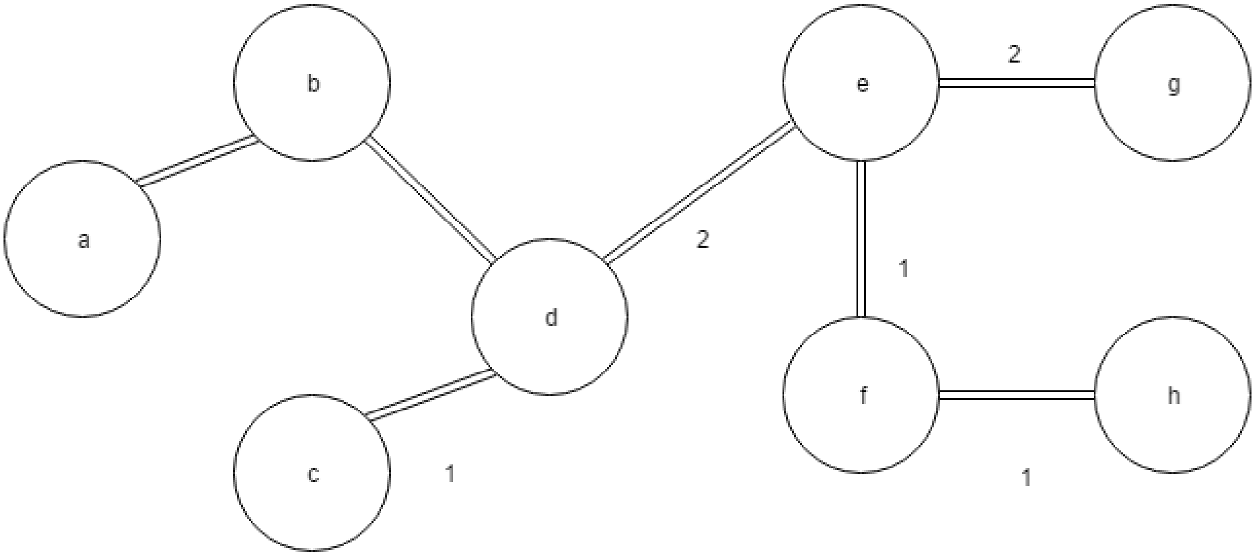
This will have a linear run time with the number of items in the adjacency matrix since we have to loop through it.

4)

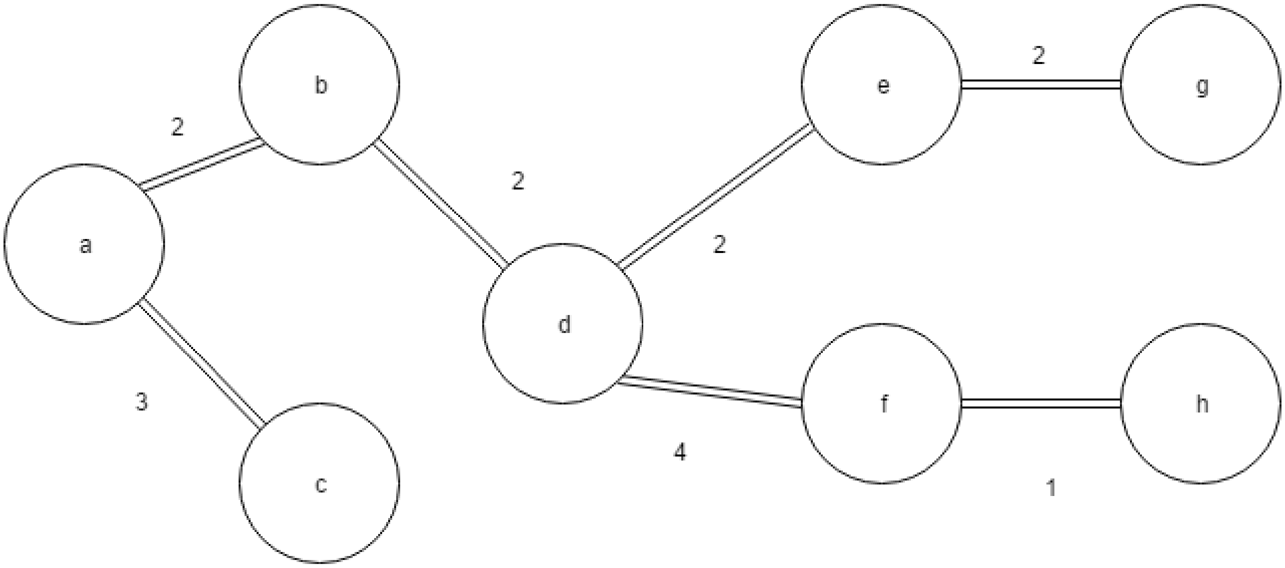
a)



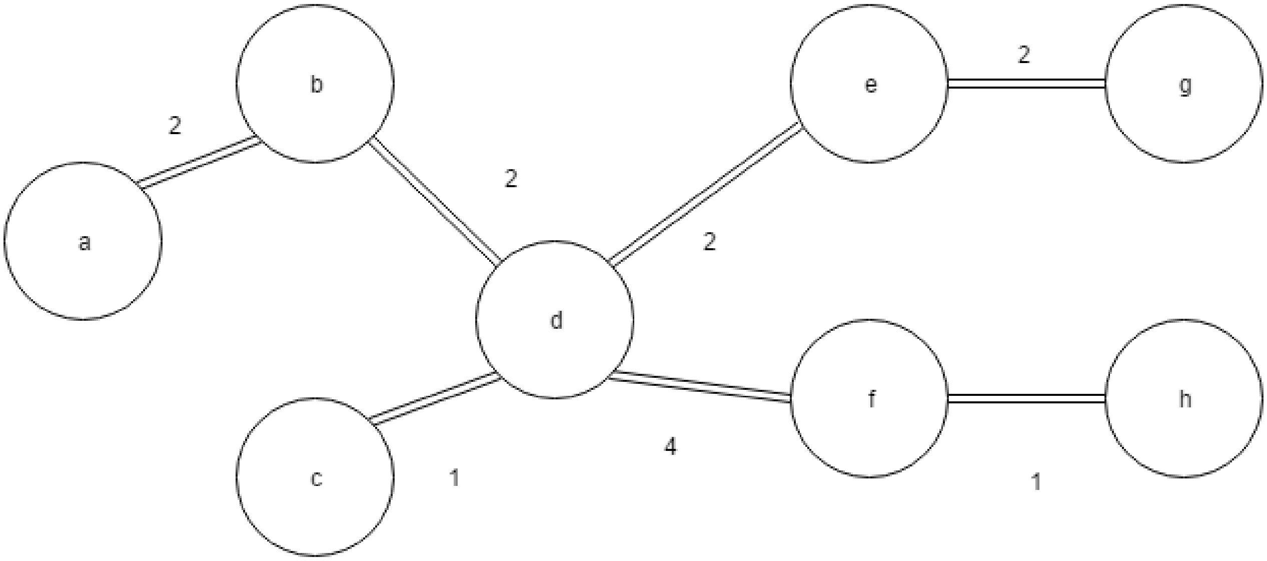
b)



d) i)



ii)



a
b
c
d
e
f
g
h
2
2
2
2
1
4
1