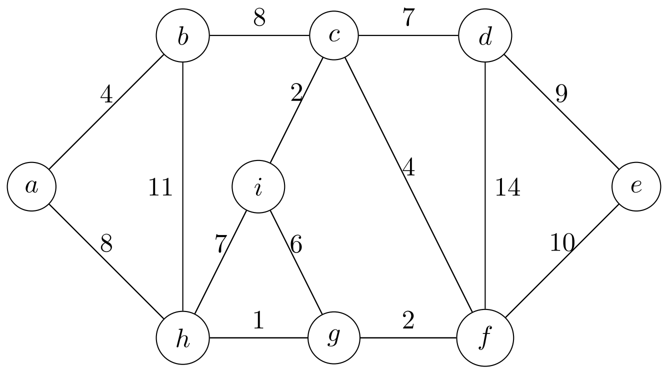
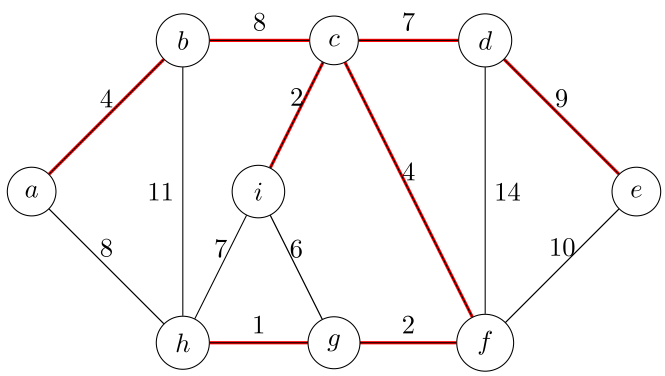
DSC 40B - Discussion 10

Problem 1.

Compute the minimum spanning tree for the following graph using Kruskal's algorithm. (Also compute the MST using Prim's algorithm and compare the results.)



Solution:



Problem 2.

Suppose we are given both an undirected graph G with weighted edges and a minimum spanning tree T of G.

a) Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge e in T is decreased.

Solution: The minimum spanning tree of the updated graph would be T.

b) Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge e not in T is increased.

> **Solution:** The minimum spanning tree of the updated graph would be T.

c) Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge e in T is increased.

> **Solution:** Let e = (u,v) be the edge whose weight is increased. Remove the edge e from the minimum spanning tree T. This divides the tree T into two connected components. Let T_u be the component which contains u and T_v be the component which contains v. We can identify T_u and T_v by running a BFS with u and v as the sources. This takes time O(V + E). While running BFS we can also label each node as 0 if it is a part of T_u and 1 if it is a part of T_v. We can now examine each edge e in the graph and find the minimum weight edge which connects a node labelled 0 to a node labelled 1. This takes time O(E). We can then add this edge to T to get the minimum spanning tree of the updated graph. The total time complexity is O(V + E).

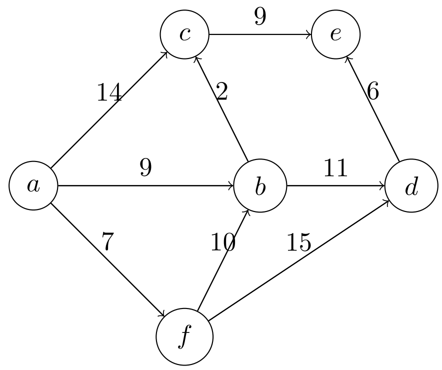
d) Describe an efficient algorithm to update the minimum spanning tree when the weight of one edge e not in T is decreased.

> **Solution:** Let e= (u,v) be the edge whose weight is decreased. Add the edge e to the minimum spanning tree T. This would result in a cycle in T. We can identify the nodes and the edges on the cycle by running BFS with u or v as the source. This takes time O(V + E). Find the maximum weight edge on the cycle and remove it from T. This takes time O(E). The resultant tree would be a minimum spanning tree for the updated graph. The total time complexity is O(V +E).

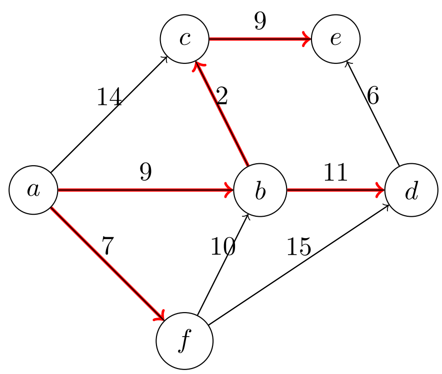
Extra Practice

Problem 3.

Run Dijkstra's Algorithm on the following graph using node a as the source. Below each node u, write the shortest path length from a to u. Mark the predecessor of u by highlighting it or making a bold arrow.



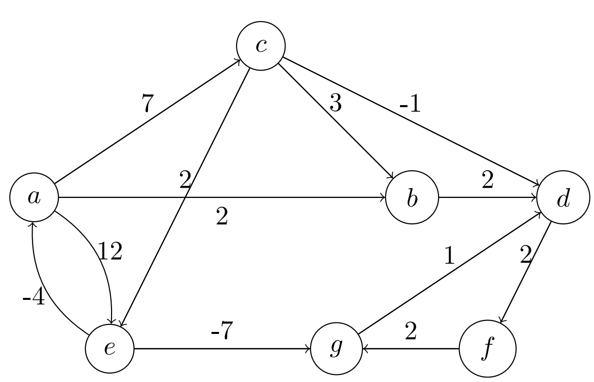
Solution:



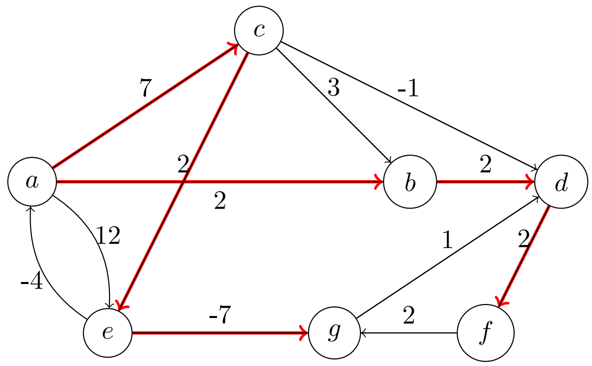
est = { 'a': 0, 'b': 9, 'c': 11, 'd': 20. 'e': 20, 'f': 7 }

Problem 4.

a) Run Dijkstra's Algorithm on the following graph using node a as the source. Below each node u, write the shortest path length from a to u. Mark the predecessor of u by highlighting it or making a bold arrow.



Solution:



est = { 'a': 0, 'b': 2, 'c': 7, 'd': 4. 'e': 9, 'f': 6, 'g': 2 }

b) Dijkstra's algorithm found the wrong path to some of the vertices. For just the vertices wherethe wrong path was computed, indicate both the path that was computed and the correct path.

Solution:
Computed path to d is a,b,d but shortest path is a,c,e,g,d.
Computed path to f is a,b,d,f but shortest path is a,c,e,g,d,f.

c) What single edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?

Solution: (e,g)

Problem 5.

True or False. Suppose $G = (V, E, \omega)$ is a weighted graph for which all edges are positive, except for those edges of a node s which may or may not be negative. If Dijkstra's algorithm is run on G with s as the source, the correct shortest paths will be found. Assume that the graph does not have any negative loops. Justify your answer.

Solution: True. The reason that Dijkstra's algorithm can fail in the presence of negative edges is that it cannot rule out that a negative edge that it has not seen will cause a path which is currently longer to eventually become shorter. But if all of the negative edges are attached to the source, the rest of the edges are still positive; and so there is no way in which adding edges to a path can decrease its length.