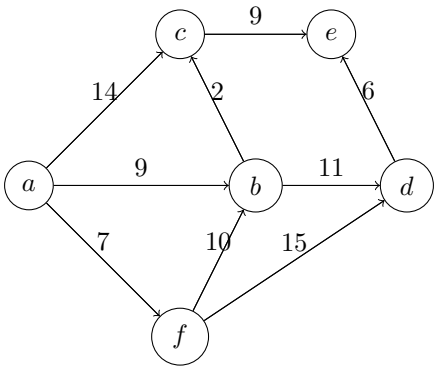
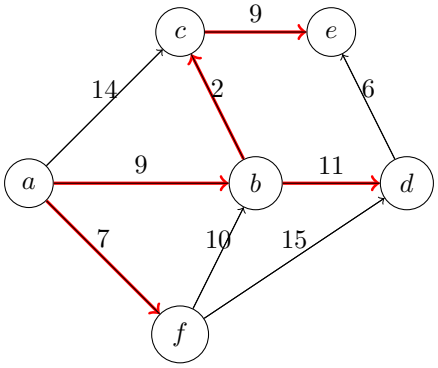
DSC 40B - Discussion 09

Problem 1.

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.



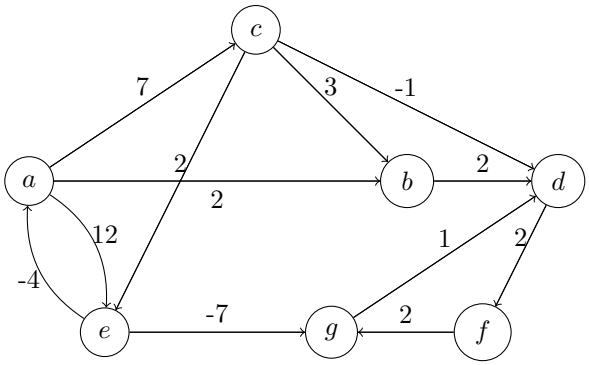
Solution:



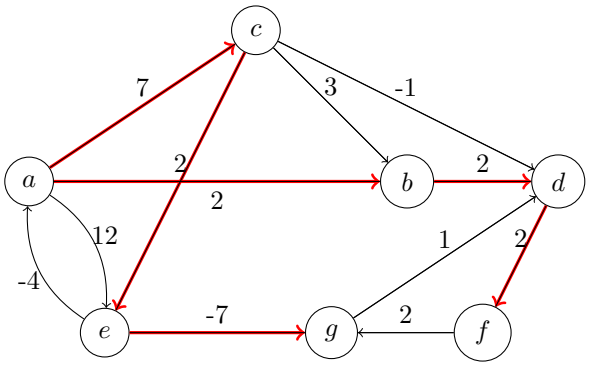
est = { 'a': 0, 'b': 9, 'c': 11, 'd': 20. 'e': 20, 'f': 7 }

Problem 2.

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.

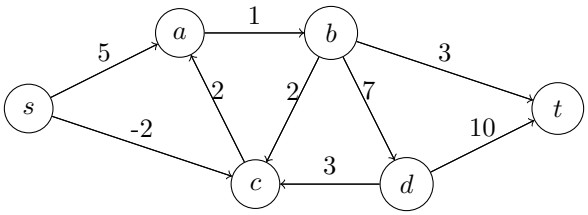
c) What single edge could be removed from the graph such that Dijkstra's algorithm would happen to compute correct answers for all vertices in the remaining graph?

Solution: (e,g)

Problem 3.

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

```
def bellman_ford(graph, weights, source):
    est={node:float('inf') for node in graph.nodes}
    est[source]=0
    predecessor={node: None for node in graph.nodes}
    for i in range(len(graph.nodes)-1):
        for(u, v) in graph.edges:
            update(u, v, weights, est, predecessor)
    return est, predecessor
```



Solution:
predecessor : {'s':None, 'a':'c', 'b':'a', 'c':'s','d':'b', 't':'b'}
est : {'s':0, 'a':0, 'b':1, 'c':-2, 'd':8', 't':4}