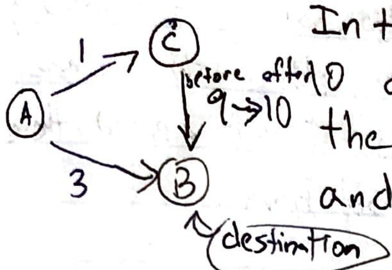
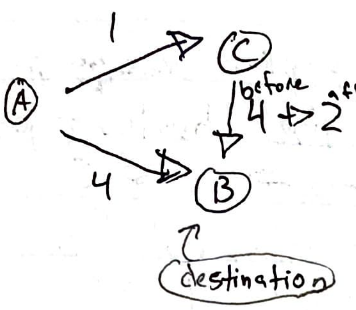
Gordon Ng Homework 5

①



In this case, changing the edge weight from 9 to 10 doesn't change the distance of the nodes, as the edge weight from node A to node B is 3 and this is always smaller than 9 or 10.

Where it does:



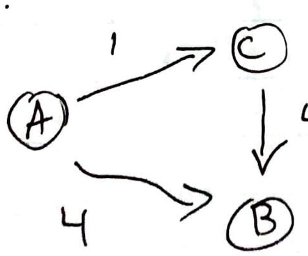
In this case, changing the edge weight from 4 to 2 does change the distance of the nodes, as the edge weight from node A to B was less than/~~equal~~ to the node weights of A->C->B. However, once 4 becomes 2, the ~~node~~ edge weight from A->B becomes greater than the edge weights from A->C->B.

② Decrease Edge Weight (dist, T):

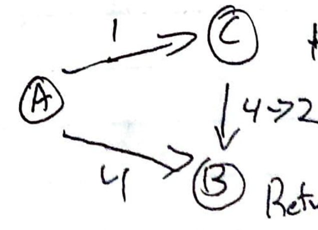
```
    if ((dist [u] + w'(u,v)) < dist[v]):
        return yes
    else:
        return no
```

Dijkstra's gives us the shortest path, because of this, we need to find if the changed edge node makes the shortest path even shorter. This algorithm takes the given changed node and adds it to the path before it. It then compares if the segment is less than the shortest path. If it is less than the shortest path, then there is a new shortest path in town. If it is not less/shorter than the shortest path, then the shortest path remains the same and it should return no. Due to the one if conditional statement and that dist [u] and w'(u,v) are given to you, the retrieval of this information and the if statement doesn't exceed O(1) time. So, time complexity is O(1).

Proof:

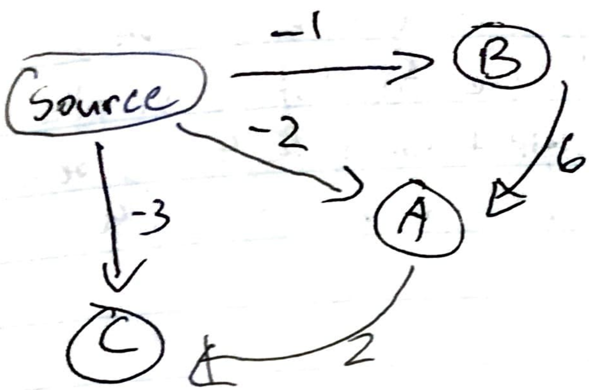


Here, dist[v] is from A->B, 4.



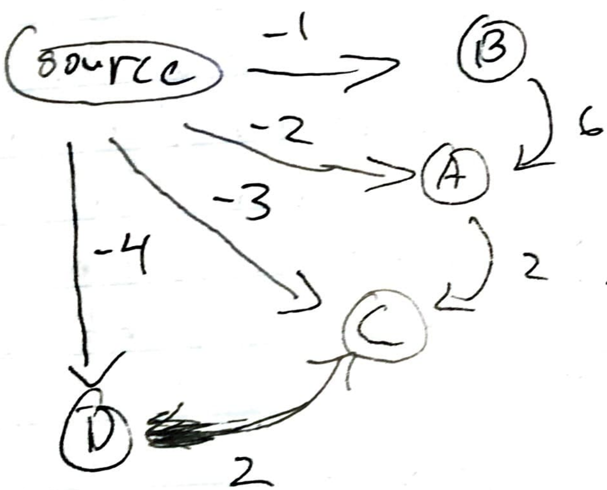
Changing 4 to 2, ~~the~~ algorithm adds A->C, 1 to C->B, 2. Returning yes, there is a change.

3



Base:

In this case the shortest path exists with Path source → C. If Dijkstra's is applied, there would be a solution.



k+1:

Dijstra's is applied once again here where the shortest path is is Path Source → D → C.

There is an extra node here.

The claim holds for $k \geq 1$