Bellman - Ford Algorithm

- It is a Dynamic Programming approach. (It will be finding out all possible solutions and gives us best answer / solution).

- Like Dijkstra, it also find the shortest path from source vertex to all other vertices of a graph.

- Dijkstra fails if there is a negative edge weight but bellman ford algorithm works for such graph.

- Here we will not be deleting the vertex with least key value, but we will be performing iterations.

  No. of iterations = No. of vertices − 1

- No deletion is required, therefore not considering minheap.

- If $d[u] + w(u,v) < d[v]$
  then $d[v] = d[u] + w(u,v)$

This is known as edge relaxation.

BELLMAN - FORD (G, w, s)

```
1. INITIALIZE - SINGLE - SOURCE (G, S)
2. for i ← 1 to |V(G)| - 1
3.     do for each edge (u,v) ∈ E[G]
4.             do RELAX (u,v,w)
5. for each edge (u,v) ∈ E[G]
6.     do if d[v] > d[u] + w(u,v)
7.             then return FALSE
8. return TRUE
```

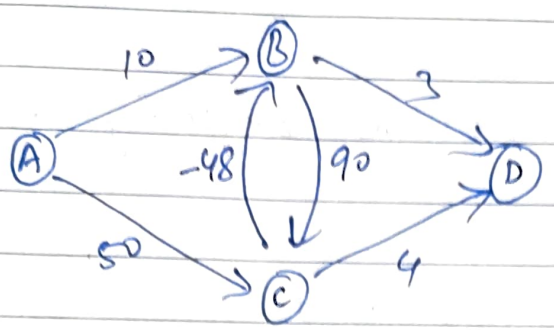
INITIALIZE - SINGLE - SOURCE (G, s)

```
1. for each vertex v ∈ V[G]
2.     do d[v] ← ∞
3.        π[v] ← NIL
4. d[s] ← 0
```

RELAX (u, v, w)

```
1. if d[v] > d[u] + w(u,v)
2.    then d[v] ← d[u] + w(u,v)
3.         π[v] ← u.
```

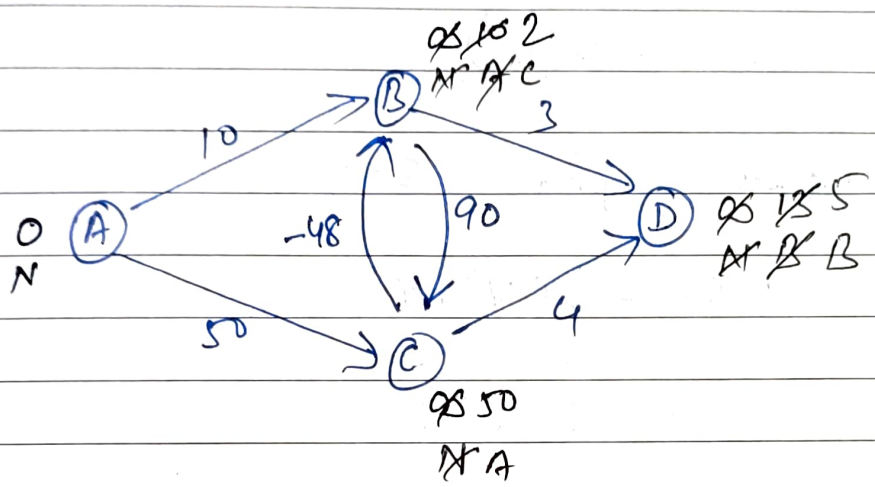
Ex:



$|V| = 4$.

∴ No. of iteration = 4-1 = 3

Edgelist: (A,B), (A,C), (B,D), (B,C), (C,D), (C,B)

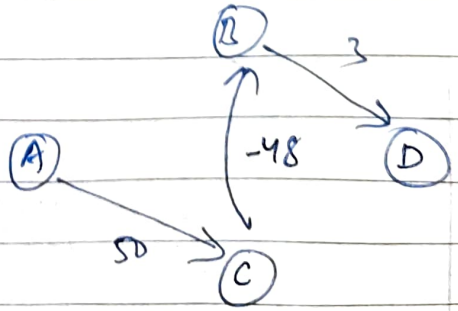


1st iteration: A : 0
B : 2
C : 50
D : 13

} Suppose we take edge-list as (B,D), (C,D), (B,C), (C,B), (A,B), (A,C).

2nd iteration: A : 0
B : 2
C : 50
D : 5

∴ solution is



3rd iteration: (same as 2nd iteration)
A : 0
B : 2
C : 50
D : 5

Time Complexity

For each edge we are performing edge relaxation V times.

$$\therefore T.C = O(E * V)$$

If E is n and V is n

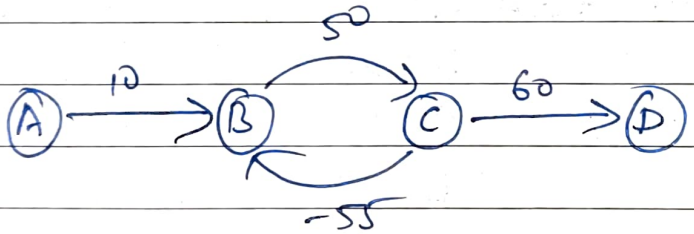
$$\therefore T.C = O(n^2)$$

For complete graph $(E = V^2)$

$$\therefore T.C = O(V^2 * V)$$

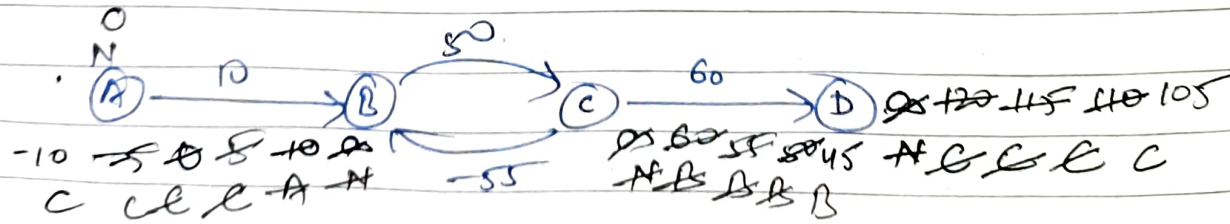
$$= O(V^3)$$

Limitations of Bellman Ford Algorithm



Sol: edge list : (A,B) (B,C) (C,D) (C,B)

$\therefore$ No. of Iterations $= 4 - 1 = 3$



1st Iteration: A : 0
B : 5
C : 60
D : 120

2nd iteration: A : 0
B : 0
C : 55
D : 115

3rd iteration: A : 0
B : -5
C : 50
D : 110

Since we have completed all iterations, let do 4th iteration just to check our answer (to see correctness of the result).

4th iteration: A : 0
B : -10
C : 45
D : 105
} values getting updated

NOTE: if in $|V|^{th}$ iteration, values changes means that there is a negative edge cycle.

i.e in $|V|^{th}$ iteration, if any value change then we can say that it will change forever.

$50 + (-55) = -5 < 0$

∴ there is no possible solution exist.

Page No.:
Date:

Imp Points:

① If graph contain all positive edge weight, then Dijkstra algorithm always give correct answer.

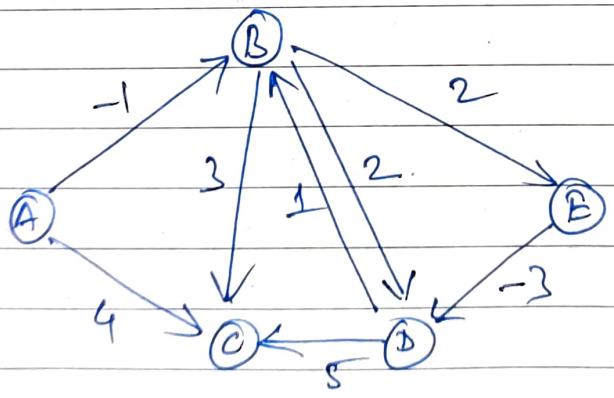
② If there is -ve edge, then Dijkstra algorithm may give wrong answer.

③ If there is a -ve edge weight cycle with sum of edge cost is +ve, then it (Bellman Ford Algorithm) will give right answer.

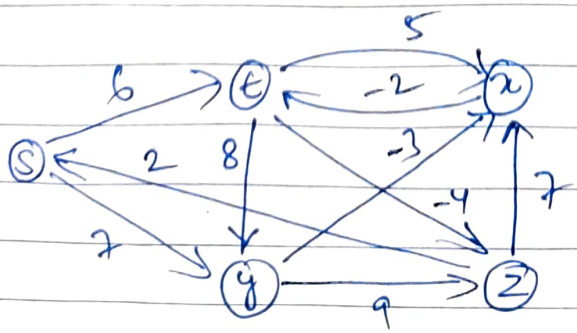
④ If there is a -ve edge weight cycle with total sum of edge cost as -ve then Bellman Ford algorithm can identify and tell that there exists -ve edge cycle.

Questions

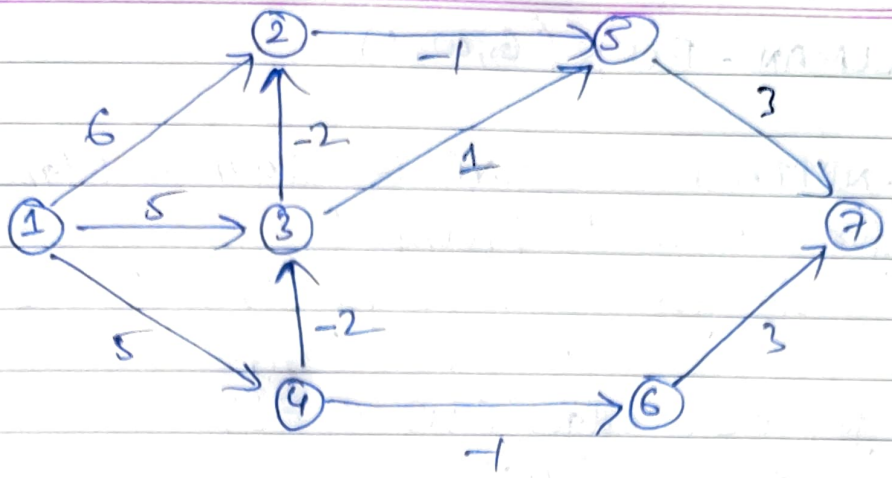
Q1



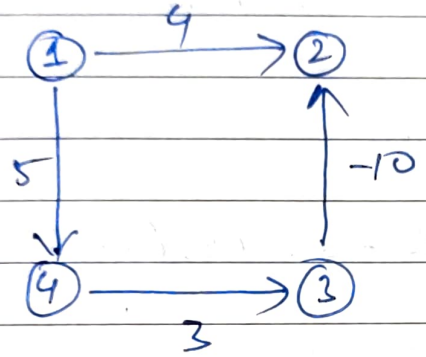
Q2



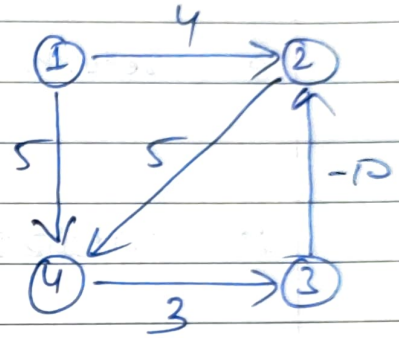
Q3



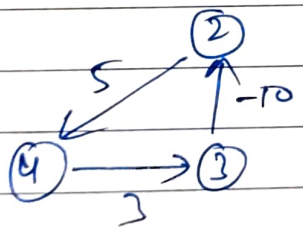
Q4



Q5



In Q5, there is a negative edge weight cycle,



$5 + 3 + (-10) = -2$