Bellman-Ford Algorithm

The Bellman-Ford algorithm solves the single-source shortest paths problem in the general case in which edge weights may be negative. If there is such a cycle, the algorithm indicates that no solution exists. If there is no such cycle, the algorithm produces the shortest paths and their weights.

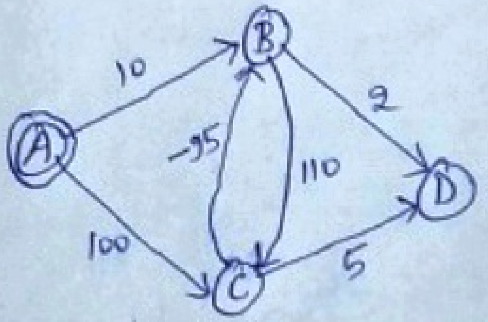
The algorithm relaxes edges, progressively decreasing an estimate v.d on the weight of a shortest path from the source s to each vertex $v \in V$ until it achieves the actual shortest-path weight $\delta(s,v)$.

```
Bellman-Ford(G, w, s)
Initialize-single-source(G, s)
For i = 1 to |G.V| - 1
For each edge (u,v) ∈ G.E
Relax(u, v, w)
For each edge (u,v) ∈ G.E
If v.d > u.d + w(u,v)
return FALSE
return TRUE
```

```
Initialize-single-source(G, s)
For each vertex v ∈ G.V
v.d = ∞
v.π = NIL
s.d = 0
```

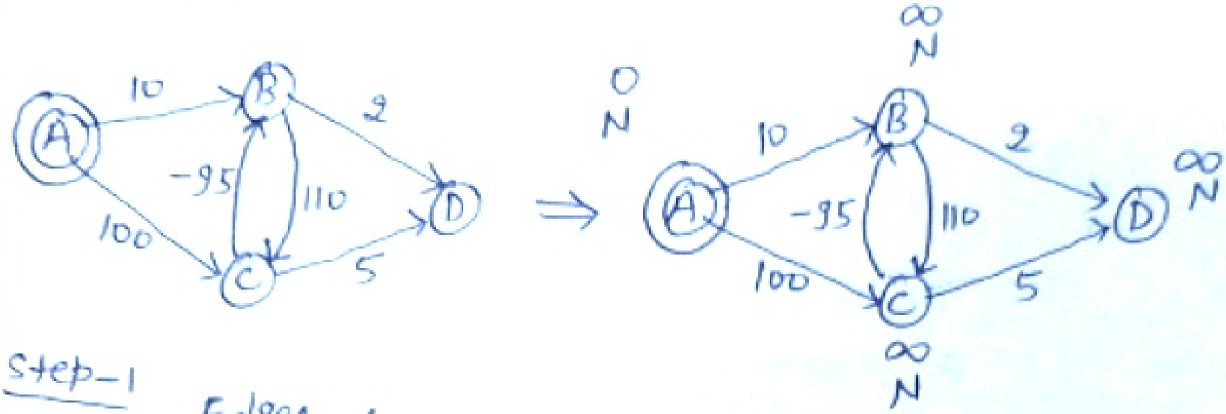
```
Relax (u, v, w)
If v.d > u.d + w(u,v)
v.d = u.d + w(u,v)
v.π = u
```

EX-1



NOTE: GO on relaxing all the edges (n-1) times.
n = # of vertices

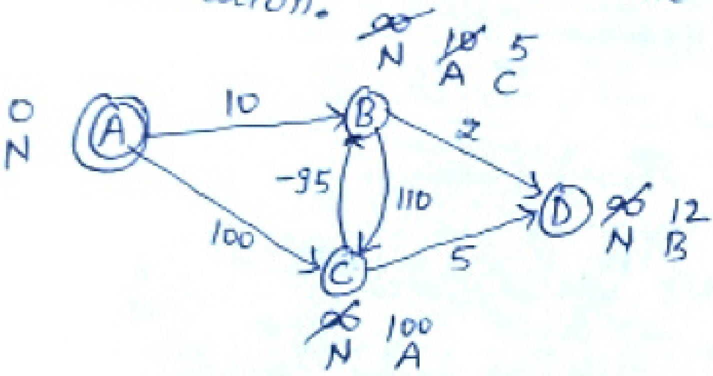
Sol – Ex-1



Step-1

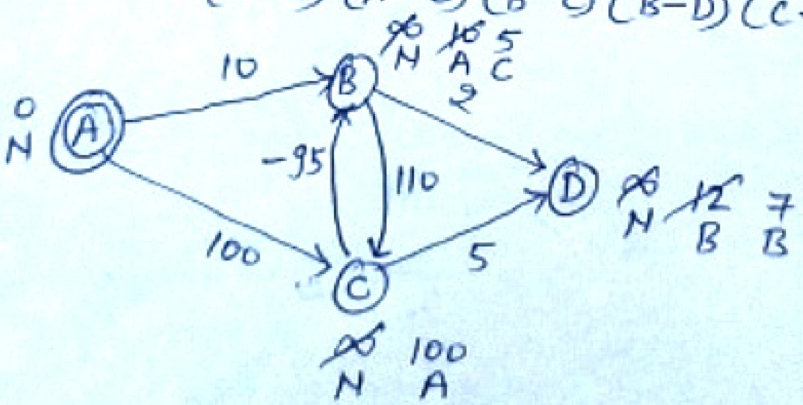
Edges (A–B) (A–C) (B–C) (C–B) (B–D) (C–D)

Extract all the edges one by one and call relax function.

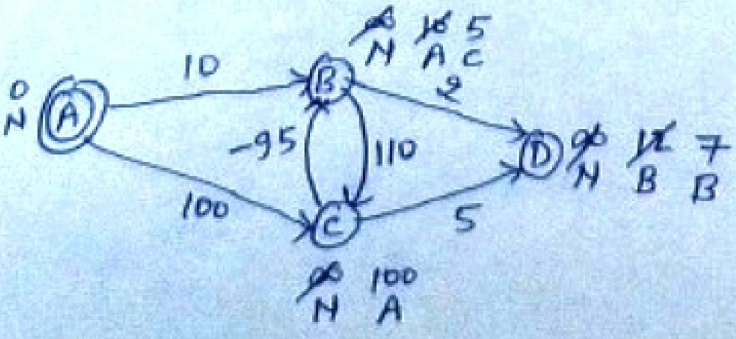


Step-2

Edges (A–B) (A–C) (B–C) (B–D) (C–B) (C–D)



Round-3



Round = V−1
TC = O(VE)
For complete graph $E = \frac{V(V-1)}{2}$ $TC = O(V^3)$