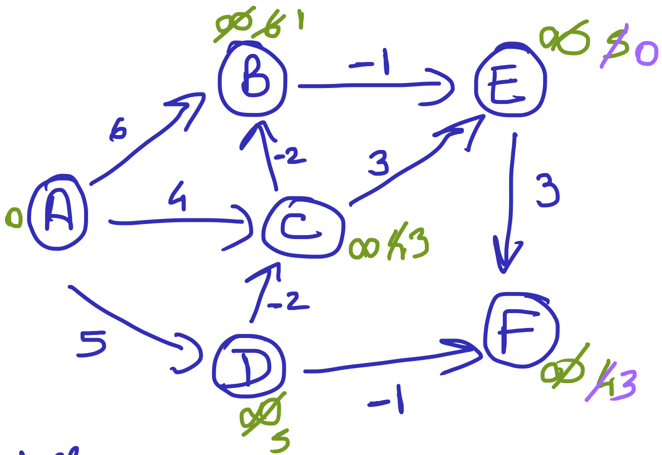
Bellman Ford Algorithm {single source shortest path}



1) go on relaxing all edges (n-1) times (n = no of vertices)

> if $d[u] + c(u,v) < d[v]$
>
> $d[v] = d[u] + c(u,v)$

edges
(A,B) (A,C) (A,D) (B,E) (C,E) (D,C) (D,F) (E,F) (C,B)

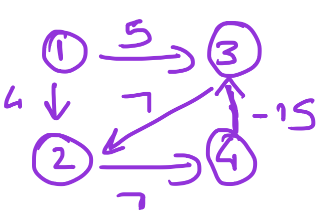
1st ✓
2nd ✓
3rd ✓ done
4th
5th

A - 0
B - 1
C - 3
D - 5
E - 0
F - 3

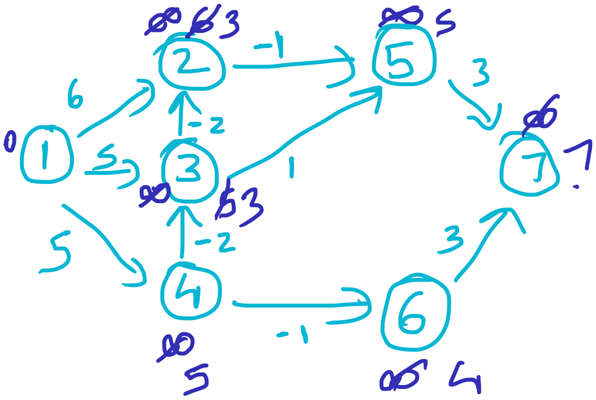
> $O(E \cdot |V|-1)$
>
> $O(n^2)$

> For complete graph, $O\left(\frac{n(n-1)}{2} \cdot (n-1)\right) = O(n^3)$

Drawback - does not work if there is a cycle having -ve weight



→ should have 3 iterations, but ends up having more



edge list ⇒ (1,2) (1,3) (1,4) (2,5) (3,2)
(3,5) (4,3) (4,6) (5,7) (6,7)

```
func BELLMANFORD(G, s):
    for v ∈ G:
        d[v] = ∞
        π[v] = Null
    d[s] = 0
    for i = 1 to |V| - 1:
        for (u,v) ∈ E:
            newdist = d[u] + w(u,v)
            if newdist < [v]:
                d[v] = newdist
                π[v] = u
    for (u,v) ∈ E
        if d[v] > d[u] + w(u v):
            return False
    return d
```

$\boxed{O(mn)}$