SHORTEST PATH ALGORITHMS (All these algos are applicable on directed & weighted graphs only)

→ Single Source Shortest Path
- Dijkstra's Algorithm
- Bellman Ford Algorithm

[When a shortest path is calculated from a single vertex (node) to all other vertices in the graph.]

→ Multiple Source Shortest Path
- Floyd Warshall Algo

[When a shortest path is calculated b/w any pair of vertices in the graph.]

1) DIJKSTRA'S ALGO - Can work on directed as well as non-directed graph

∞ → initial

$1 \xrightarrow{2} 2 \xrightarrow{4} 3$

u = 2, v = 3

6 → final

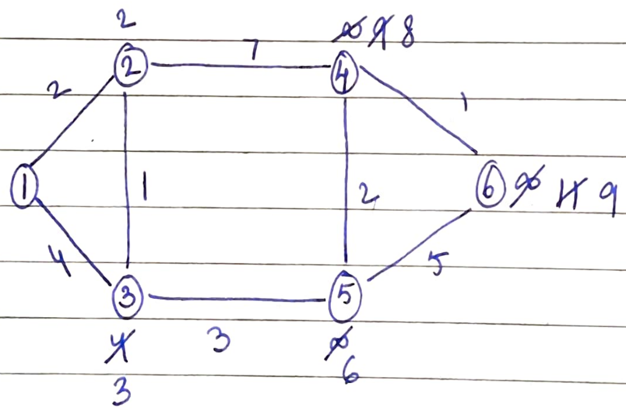
Relaxation:

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

$d[u] = 2 \quad c(u,v) = 4$

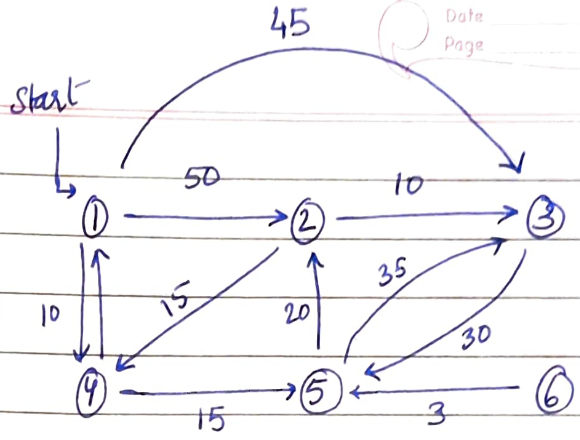
$d[v] = 6 < \infty$

Start →



V	d[V]
2	2
3	3
4	8
5	6
6	9

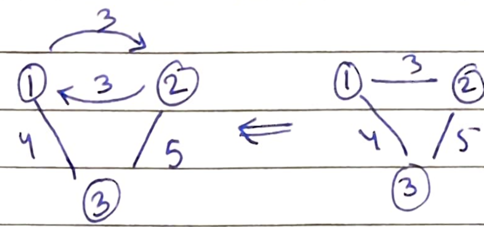
Worst Case T.C. : $O(|V|^2)$



Selected vertex	2	3	4	5	6
4	50	45	10	∞	∞
5	50	45	10	25	∞
2	45	45	10	25	∞
3	45	45	10	25	∞
6	45	45	10	25	∞

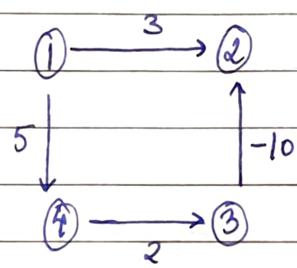
Min. val. at 4 so, 4 is selected

Convert non-directed to directed



★ No need to check vertices which are already relaxed (circled)

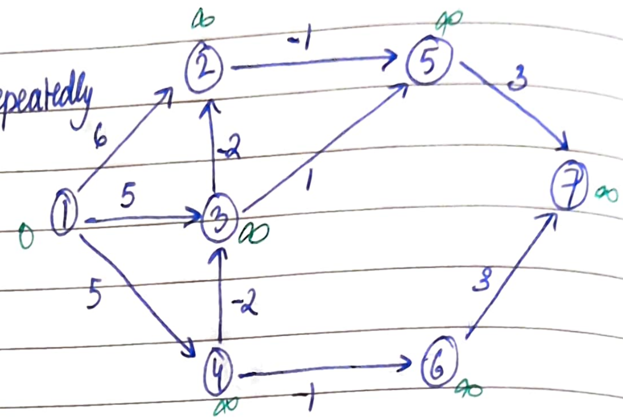
<u>Drawback</u>: Negative weighted edges (May or May not give correct ans)



Selected vertex	2	3	4
2	3	∞	5
4	3	∞	5
3	3	7	5
	-3		

↳ but Dijkstra's algorithm told us NOT to check since it is already relaxed.

2) <u>BELLMAN FORD ALGO</u>



★ Go on relaxing all edges repeatedly ~~for~~ for n-1 times.

$|V| = n = 7$

$|V|-1 = n-1 = 6$

★ Initially take start as 0 and mark dist. for all other vertices as ∞.

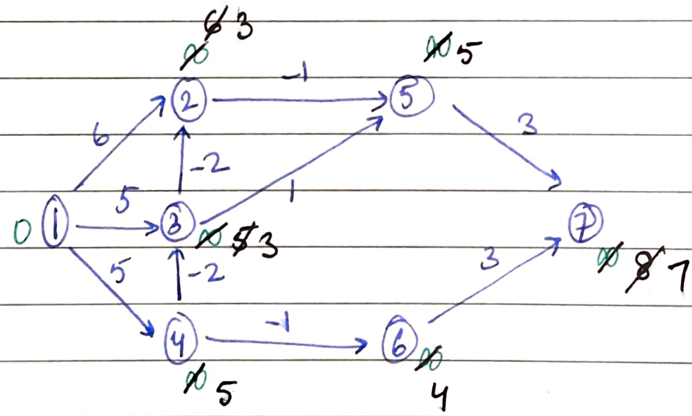
if there is an edge b/w u & v:-

<u>Relaxation</u>

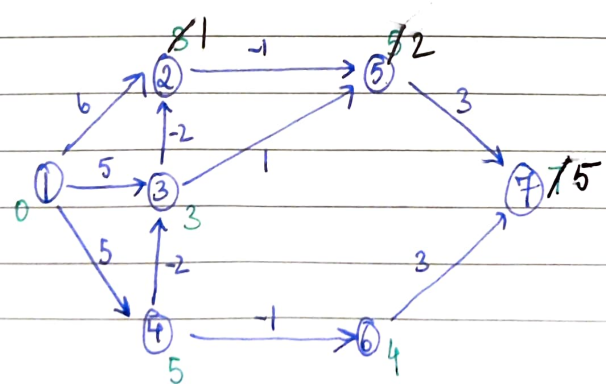
if $(d[u] + c(u,v) < d[v])\{$
$d[v] = d[u] + c(u,v)\}$

Edgelist :
(1,2), (1,3), (1,4), (2,5), (3,2), (3,5), (4,3), (4,6), (5,7), (6,7)

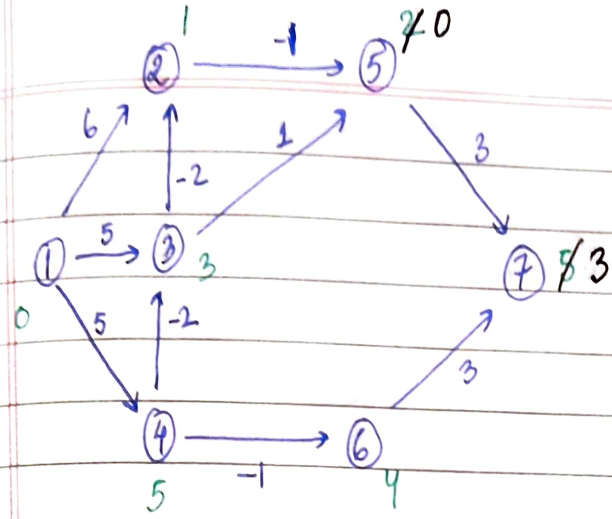
1st time :



2nd time :



3rd time:

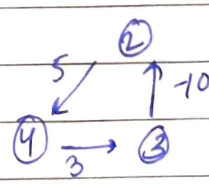


V	d[V]
1	0
2	1
3	3
4	5
5	0
6	4
7	3

4th time: NO CHANGE

Min^m T.C. $O(n^2)$

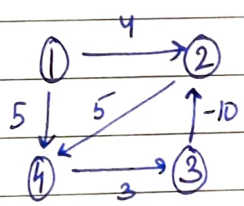
Max^m T.C. $O(n^3)$



$5+3-10=-2$

-ve weighted cycle

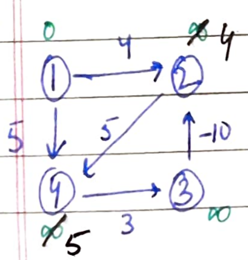
Drawback : Fails at Negative weighted cycle



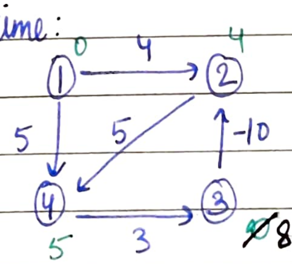
We need to perform relaxation |V|-1 i.e. 3 times.

Edge List : (3,2), (4,3), (1,4), (1,2), (2,4)

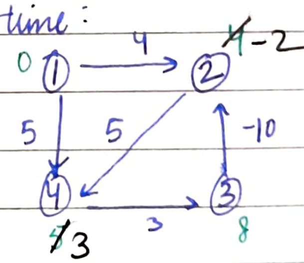
1st time:



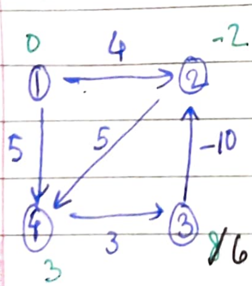
2nd time:



3rd time:



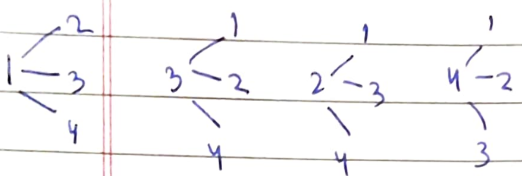
Let's try 4th time:

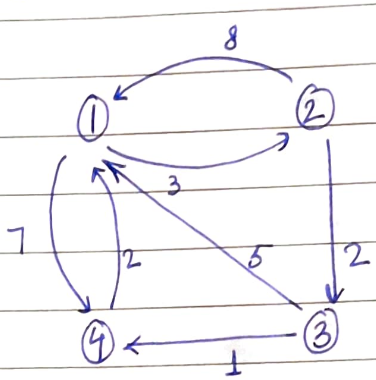


One more Vertex relaxed after n-1 times!

3) <u>FLOYD WARSHALL ALGO</u> → +ve or -ve edge weights (but NO -ve cycles)

↳ Shortest path b/w every pair of vertices





$$A^0 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & \infty \\ 5 & \infty & 0 & 1 \\ 2 & \infty & \infty & 0 \end{bmatrix} \end{array}$$

DP- We take a series of decisions at each step to get to the solution.

To find if there is any better/shorter path going via vertex 1, we make a matrix

↳ keep rows & columns against 1 as same as in A^0

→ Also fill diagonals as 0 since there are no self loops.

$$A^1 = \begin{array}{c} \\ 1 \\ 2 \\ 3 \\ 4 \end{array} \begin{array}{c} \begin{array}{cccc} 1 & 2 & 3 & 4 \end{array} \\ \begin{bmatrix} 0 & 3 & \infty & 7 \\ 8 & 0 & 2 & 15 \\ 5 & 8 & 0 & 1 \\ 2 & 5 & \infty & 0 \end{bmatrix} \end{array}$$

$A^0[2,3]$ $A^0[2,1] + A^0[1,3]$

$2 < 8 + \infty$

$\therefore A^1[2,3] = 2$

$A^0[2,4]$ $A^0[2,1] + A^0[1,4]$

$\infty > 8 + 7$

$\therefore A^1[2,4] = 15$

Similarly, do for all

→ Keep rows and columns against 2 as same as in A^1

→ Diagonals also remain 0.

$$\begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 7 \\ 2 & 8 & 0 & 2 & 15 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array} = A^2$$

$A'[1,3] \quad A'[1,2]+A'[2,3]$

$\infty \quad > \quad 3+2$

$\therefore A^2[1,3]=5$

$A'[1,4] \quad A'[1,2]+A'[2,4]$

$$A^3 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 7 & 0 & 2 & 3 \\ 3 & 5 & 8 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

$$A^4 = \begin{array}{c|cccc} & 1 & 2 & 3 & 4 \\ \hline 1 & 0 & 3 & 5 & 6 \\ 2 & 5 & 0 & 2 & 3 \\ 3 & 3 & 6 & 0 & 1 \\ 4 & 2 & 5 & 7 & 0 \end{array}$$

⇒ Shortest path b/w ~~all~~ every pair of vertices

FORMULA: $\boxed{A^k[i,j] = \min\{A^{k-1}[i,j],\ A^{k-1}[i,k] + A^{k-1}[k,j]\}}$