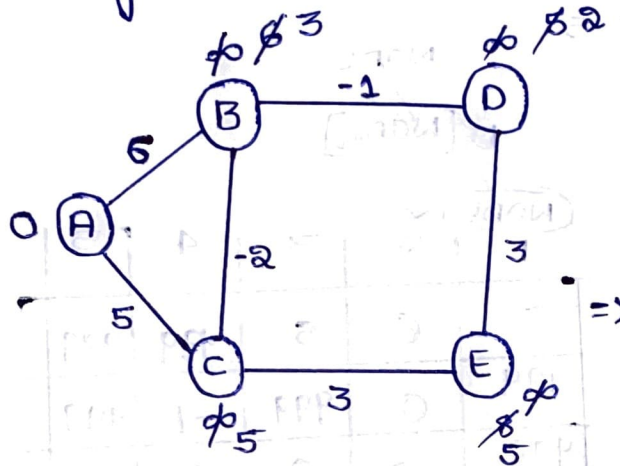


Bellman Algorithm :

↳ Bellman ford algorithm is used to find shortest path from source to all vertices in the given graph

↳ Dijkstra's Algo doesn't work for negative weight edges.



5 vertices = (5-1)
= 4(iteration)

$\Rightarrow (A,B) (A,C) (B,D)$
 $(C,B) (C,D) (C,E) (D,E)$

1st Iteration

$$0 + 6 < \infty$$

$$0 + 5 < \infty$$

$$6 - 1 < \infty$$

$$5 - 2 < \infty$$

$$5 + 3 < \infty$$

2nd iteration

$$3 - 1 < 5$$

$$2 + 3 < \infty$$

3rd and 4th iteration

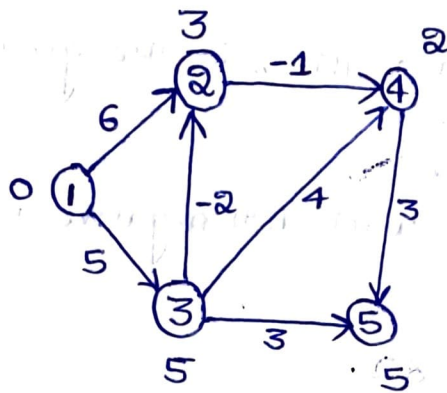
same as 2nd iteration

\Rightarrow iteration done through formula like

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

$$d[v] = d[u] + c[u,v]$$

In system instead of Alphabets
we given numbers



NODE	0	1	2	3	4	5
D[NODE]		0	999	999	999	999

before

~~NODE~~
~~D[NODE]~~

D[NODE] →

NODE ↓	1	2	3	4	5
0		6	5	999	999
1	999				
2	999	0	999	-1	999
3	999	-2	0	4	3
4	999	999	999	0	3
5	999	999	999	999	0

* In system we will give zero instead of 999

↳ because there is no edge going back to the same vertices in above figure.

* Trick to remember crack the code *

Step-1 : idata

Step-2 : bellman evaluation

Step-3 : formula

Step-4 : negative edge code stops

Step-5 : Display the distance of vertices