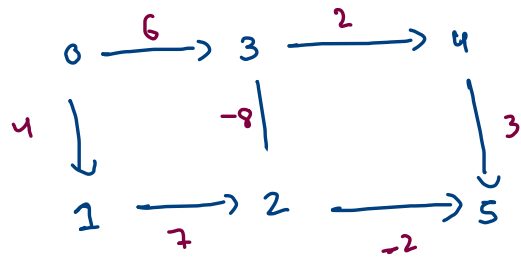


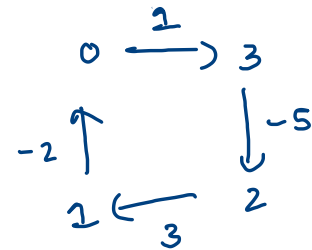
## BellmanFord :

- (i) single src all dest shortest path (edge wt).
- (ii) DP based algo.
- (iii) it works on -ve edge wt.
- (iv) it detects -ve wt cycle.

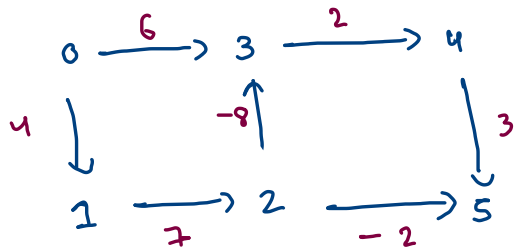


src = 0

-ve wt cycle  
(shortest path  $\rightarrow$  meaningless)



src = 0



src  $\rightarrow$  0

$i^{th}$  iteration:  
(1 based itr)

0	<del>4</del>	<del><math>\infty</math></del>	<del>6</del>	<del>8</del>	$\infty$
0	1	2	3	4	5

$(V-1)$  iterations

u v wt

4 5 3

3 4 2

2 5 -2

2 3 -8

1 2 7

0 3 6

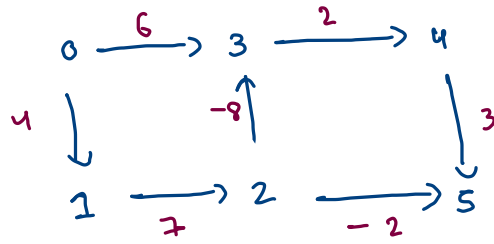
0 1 4

$u \rightarrow v$

$\text{cost}[u] + \text{wt} < \text{cost}[v]$

0 to u u to v 0 to v

5 iterations



(V-1) iterations

u v wt

4 5 3

3 4 2

2 5 -2

2 3 -8

1 2 7

0 3 6

0 1 4

u → v

cost[u] + wt < cost[v]

0 to u u to v 0 to v

0	<del>4</del>	<del>11</del>	<del>3</del> 6	<del>5</del> 8	<del>9</del> 11
0	1	2	3	4	5

0-1 shortest path: 1

0-3 shortest path: 3

0-2 shortest path: 2

0-4 shortest path: 4

0-5 shortest path: 5

ans

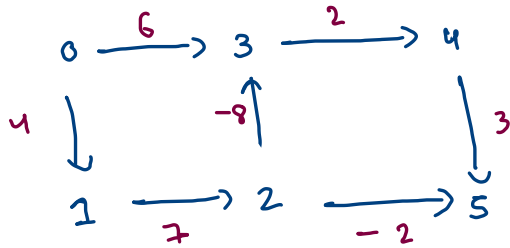
4

3

11

5

8



vertices with shortest path of  
length 'i' from src will get  
their ans in atmost i iterations  
i.e  $\leq i^{\text{th}}$  iteration

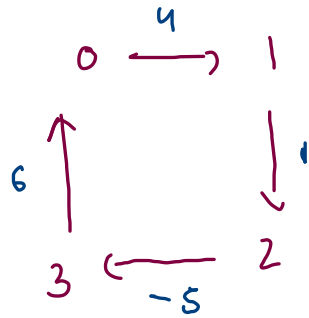
0	<del>4</del>	<del>11</del>	<del>3</del> 6	<del>8</del> 5	<del>9</del> 8
0	1	2	3	4	5

u v wt

0	1	4
1	2	7
0	3	6
4	5	3
3	4	2
2	5	-2
2	3	-8

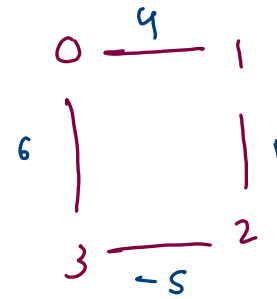
$(V-1)$

0-1 shortest path: 1      ans 4  
0-3 shortest path: 3      3  
0-2 shortest path: 2      11  
0-4 shortest path: 4      5  
0-5 shortest path: 5      8



$0 \rightarrow (0, 1, 4)$   
 $1 \rightarrow (1, 2, 1)$   
 $2 \rightarrow (2, 3, -5)$   
 $3 \rightarrow (3, 0, 6)$

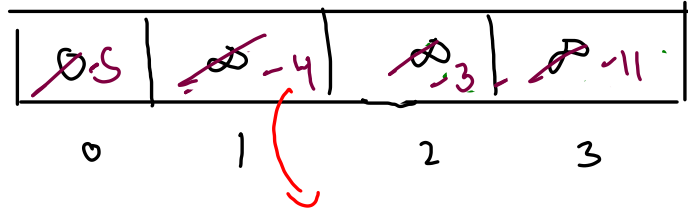
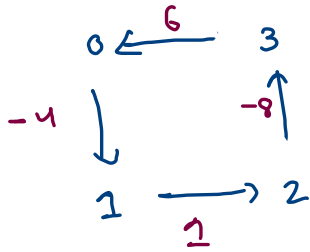
$0 \ 1 \ 4$   
 $1 \ 2 \ 1$   
 $2 \ 3 \ -5$   
 $3 \ 0 \ 6$



$u < v$

$0 \rightarrow (0, 1, 4), (0, 3, 6)$   
 $1 \rightarrow (1, 0, 4), (1, 2, 1)$   
 $2 \rightarrow (2, 1, 1), (2, 3, -5)$   
 $3 \rightarrow (3, 2, -5), (3, 0, 6)$

-ve wt cycle



$$v-1 \begin{bmatrix} 1 & 2 & 1 \\ 0 & 1 & -4 \\ 3 & 0 & 6 \\ 2 & 3 & -8 \end{bmatrix}$$

src = 0

3 iteration

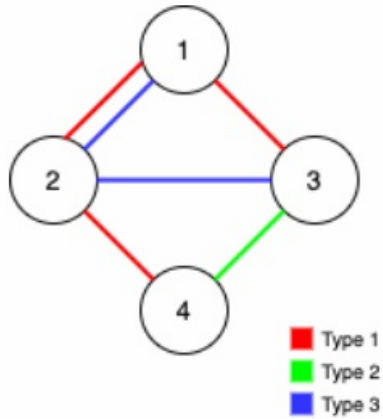
4 itx

(vth itx) updation

-ve wt cycle

Alice and Bob have an undirected graph of  $n$  nodes and 3 types of edges:

- Type 1: Can be traversed by Alice only.
- Type 2: Can be traversed by Bob only.
- Type 3: Can be traversed by both Alice and Bob.

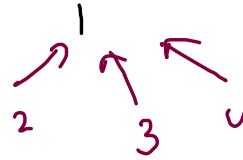


(i) priority  $\rightarrow$  type 3 edges

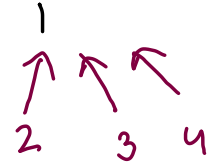
(ii) DSU

~~rem = 2~~

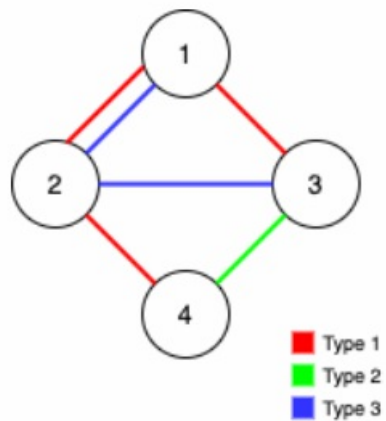
(Alice)



(Bob)



type	u	v
3	1	2
3	2	3
<del>1</del>	1	3
1	2	4
<del>1</del>	1	2
2	3	4



type	u	v
3	1	2
3	2	3
1	1	3
1	2	4
1	1	2
2	3	4

Alice	Bob