Plans - Saturday

Saturday, 30 October 2021 9:52 AM

Morning.

1. Bellman ford
2. Negative weight cycle detection
3. Max Edge Removal

Evening.

1. Colouring of Border
2. 0-1 BFS
3. Alien Dictionary
4. Min Swap to Sort an array
5. Bipartite graph.

11

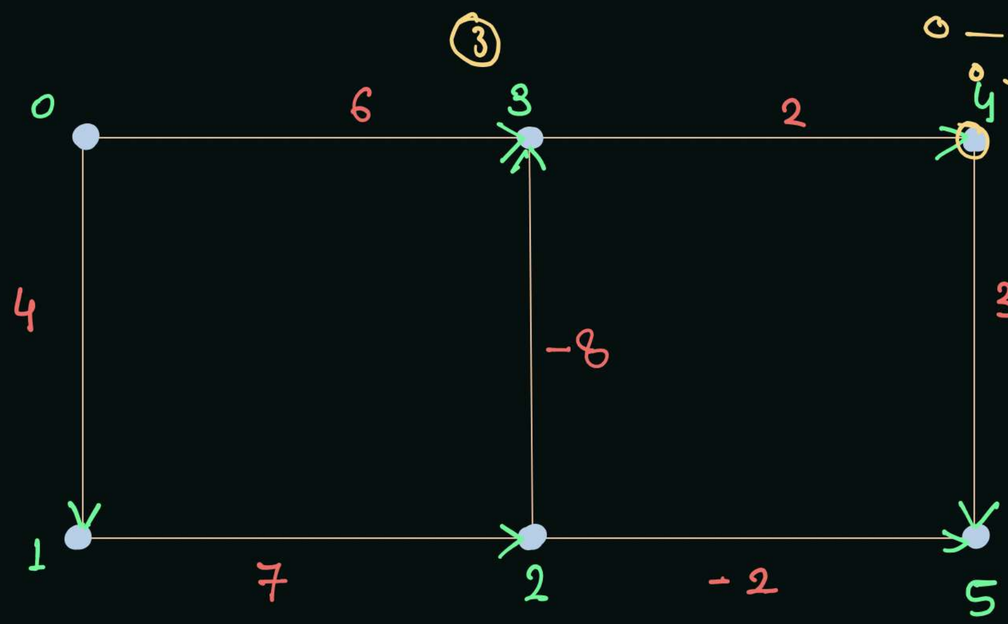
* End of graph. *

Next topic → Heap and Hashmap

Bellman Ford Algorithm

Saturday, 30 October 2021 10:14 AM

* Shortest path from single source to all destination. in terms of weights [It includes -ve weights as well].
* -ve weight cycle detection.



0 — 4 = 8
0 — 4 = 5

i=2
i=4

0	1	2	3	4	5
0	∞	∞	∞	∞	∞

P1 * Src to Src Smallest cost = 0

NOTE: Because -ve wt cycle will consider therefore Src to Src Smallest cost is 0.

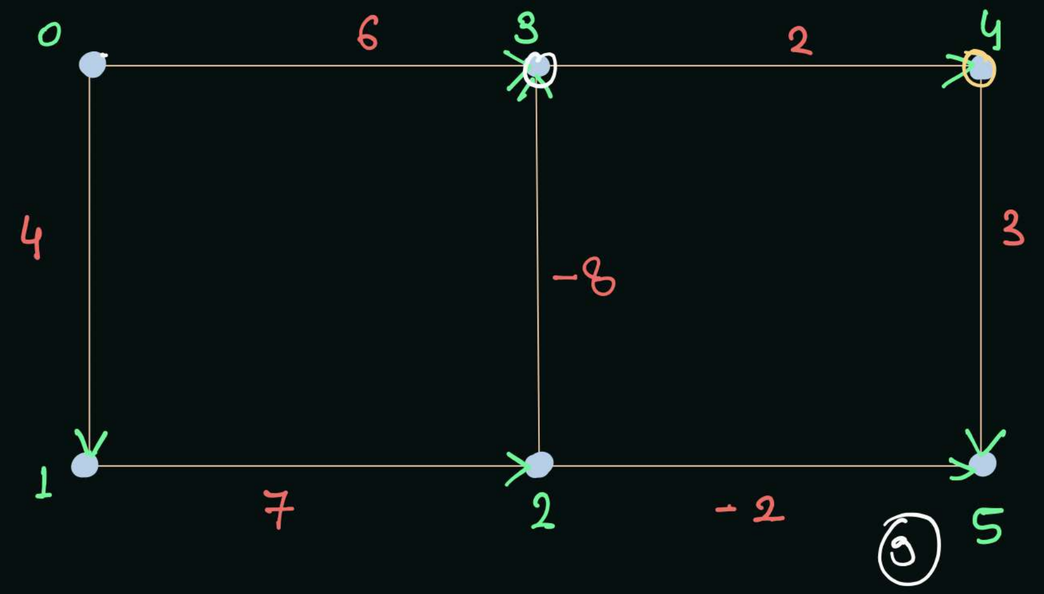
P2 * consider fix order of Edges and Solve for (V-1) time.

in i^{th} iteration shortest path length should be 'i'.

i=2

consider order of Edge ⟶

Iteration ⟶ V-1 Iteration.



u	v	wt
4	5	3
3	4	2
2	5	-2
2	3	-8
1	2	7
0	3	6
0	1	4

Iteration
1
2
3
4
5

u-v. - Edge.

```
if ( path[u] + wt < path[v])
    path[v] = path[u] + wt;
```

	0	1	2	3	4	5
path	0	4	11	3	5	8

ith path length <= ith iteration

Note: 'i' path length will definietly solve till i^{th} iteration -

why we iterate (v-1) times ??

graph

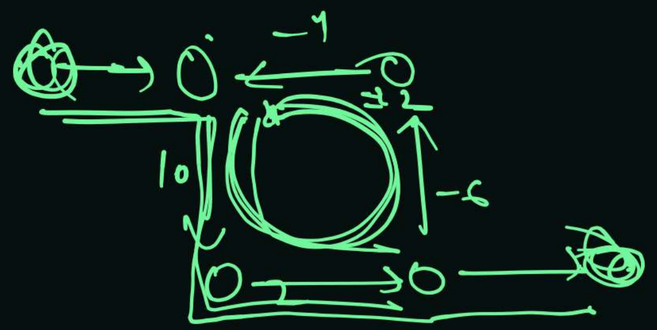
linear

V vertex

Edge → (v-1) Edge

(v-1) time iterate for path length.

cyclic



-ve wt cycle is not consider

it also take at max (v-1) distance

try to solve ⟶ (Pen paper)

0	1	2	3	4	5
0	~~∞~~	~~∞~~	~~∞~~	~~∞~~	~~∞~~
	4	11	~~6~~	5	~~9~~
			3		8

Edges:

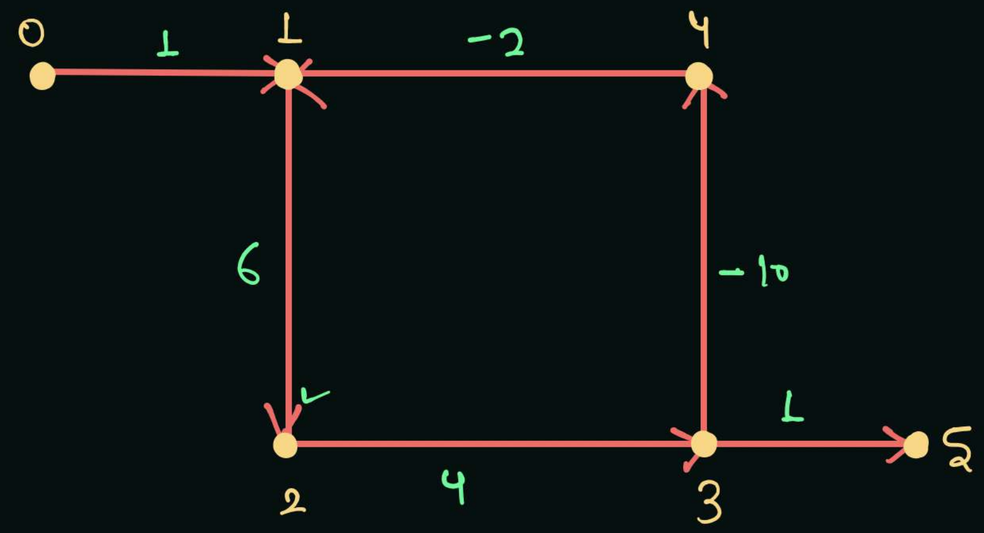
u	v	wt
0	1	4
0	3	6
1	2	7
2	3	-8
2	5	-2
3	4	2
4	5	3

1
2
3
4
5

```
if( path[u] + wt  < path[v])
    path[v] =  path[u] + wt;
```

Negative Weight Cycle Detection

Saturday, 30 October 2021 11:48 AM



(7)

1 + 6 + 1 + (-10) + (-2) + 1

= (5)

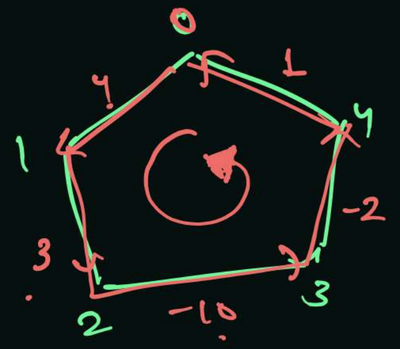
src = 0 , dst = 5

Bellman ford V = 6

5 - iterations

0 - 1 = 4

0 - 0 = 0



6th iteration

↳ Result update.

↳ -ve wt cycle exist

(4)

-3 - 2 = -5 + 1 = (-4)

Remove Max Number of Edge

Saturday, 30 October 2021 12:16 PM

Green → type - 3

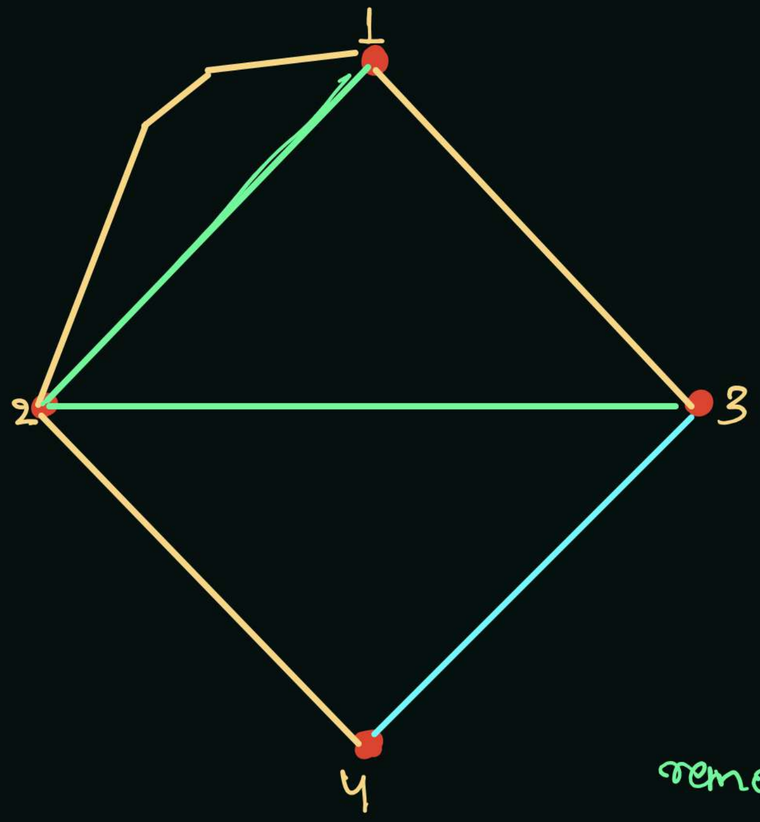
Yellow → type - 1

Skyblue → type 2

3 → Edge from Bob and Alice both can travel

1 → Edge from Alice can travel

2 → Edge from bob can travel

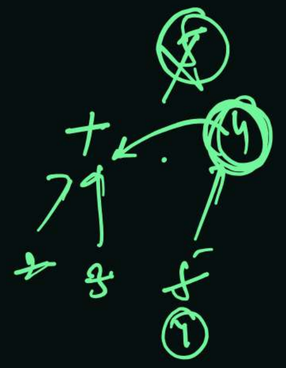


Sort on the basis of type.

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

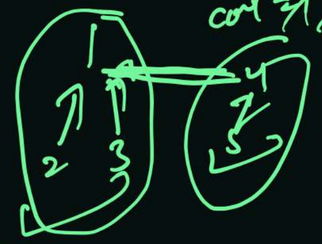
remove = ~~0~~ ~~1~~ 2 → final Result

Type	u	v
3	1	2
1	1	3
1	2	4
2	3	4
1	1	2
3	2	3

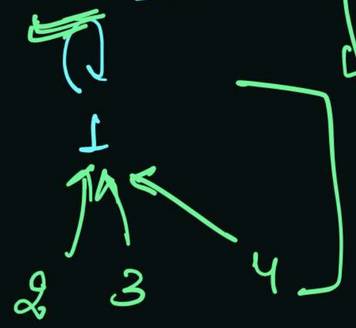


~~2~~ ~~3~~ ~~4~~ 5

cont = ~~1~~ ~~2~~ ~~3~~ ~~4~~ 5

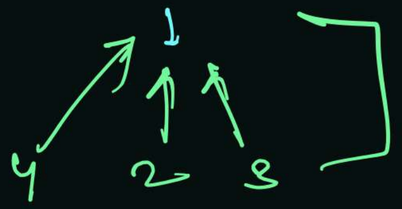


Bob



count = ~~2~~ ~~3~~ 4

Alic



count = ~~2~~ ~~3~~ 4