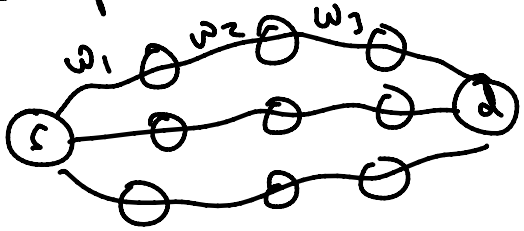
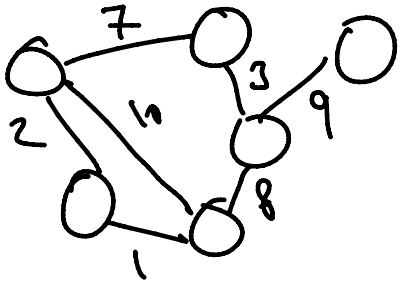
=> Shortest Path Algorithms

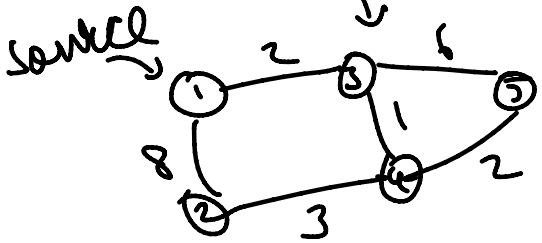
→ Given source & destination nodes, find shortest path b/w them





- Bellman-Ford
- Dijkstra Algo
- Floyd Warshall.

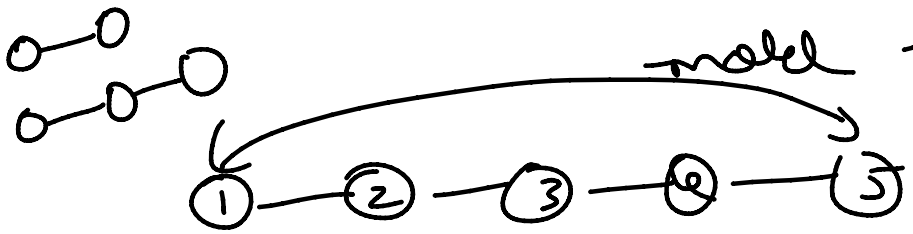
① Bellman Ford:-



→ used to find shortest distance from source to all other vertices.

→ graph → n, E

shortest path b/w 2 nodes cannot have more than $(n-1)$ edges.



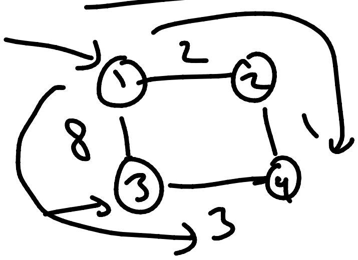
④

$E \rightarrow (n-1)$ relax.

$(\quad) \ldots < d[v]$

if(d[u] + w < d[v])
{ d[v] = d(u) + w ;

source



(n-1)

(3)

d( ) =

1	2	3	4
∞	∞	∞	∞ ~~4~~ 3
0	2	~~8~~ 6	

u	v	w
1	2	2
1	3	8
3	4	3
2	4	1

u	v	w
2	1	2
3	1	8
4	3	3
4	2	1

d[1] + 8 < d[3]
0 + 8 < ∞

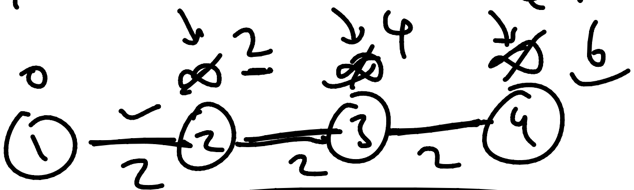
d[1] > d[2] + 2
0 > 2 + 2

d(3) > d[3] + 8

d(4) + 3 < d(3)
3 + 3 < 8

d(2) > d[4] + 1
2 > 3 + 1
2 > 4

d(2) > d[1] + 2
d(2) = 2



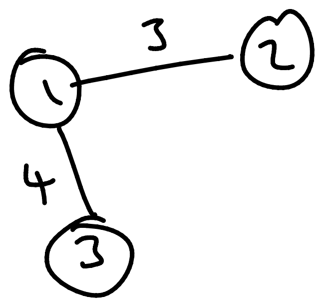
3 - 4 - 2
2 - 3 - 2
1 - 2 - 2

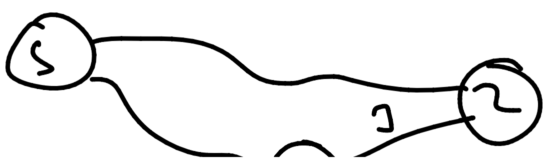
2 - 1 - 2
3 - 2 - 2
4 - 3 - 2

d(3) + 2 < d(4)
∞ + 2 < ∞

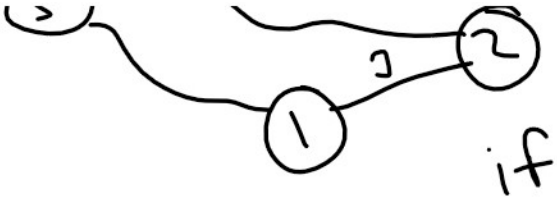
iterations = 3.

1 ⟶ (2, 3), (3, 4)
x



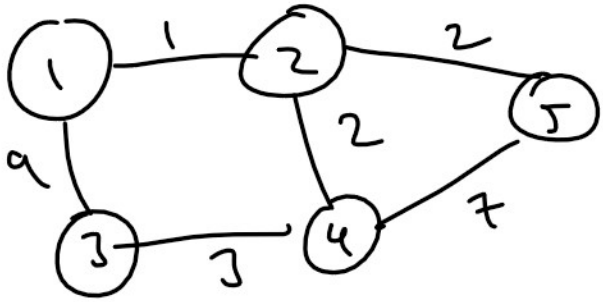


d(x.first)



if (d[i] + x.second < d[x.first])

1, 6, 3, 3, 1



→ Application : To find -ve weighted cycles.

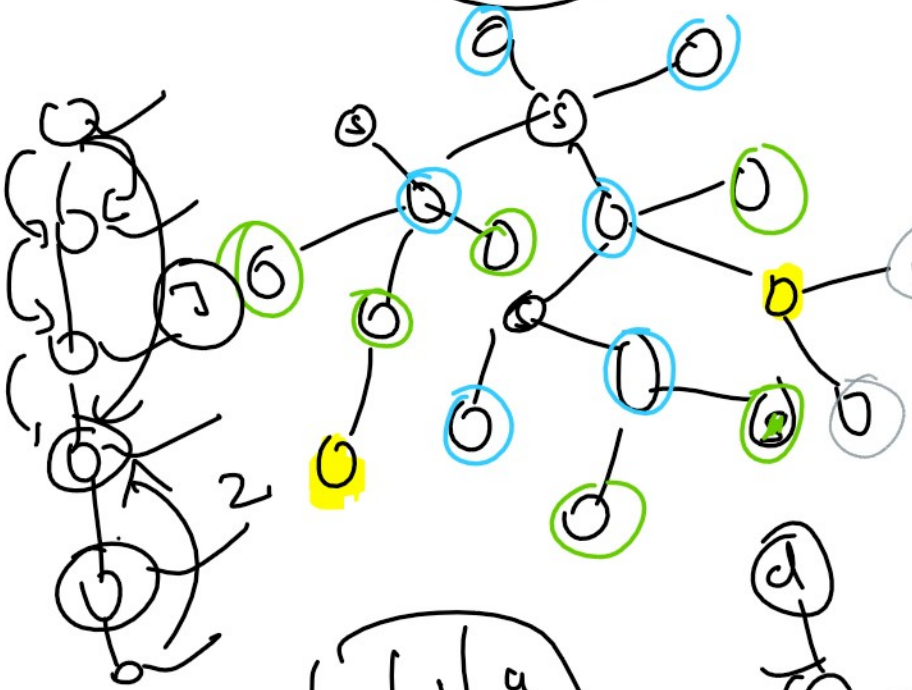
TC = O(n×e)

-1

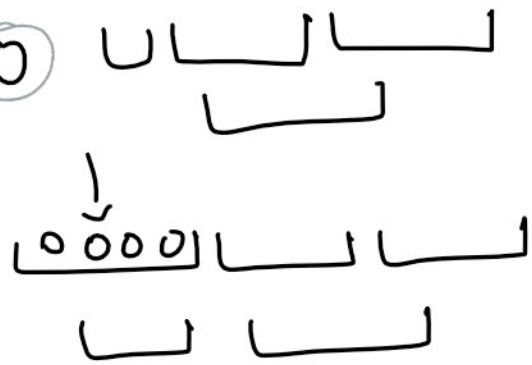
-5+3+1-5

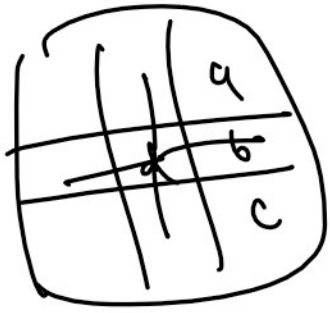
= -6

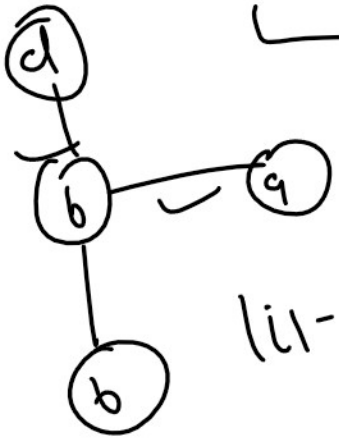
-7



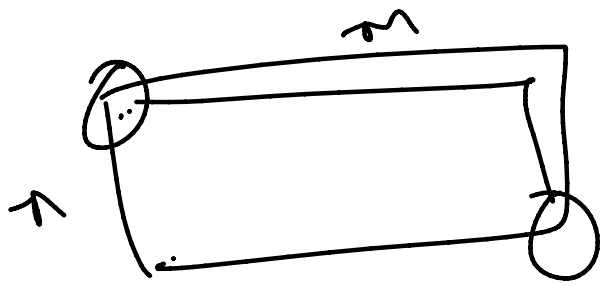
queue →





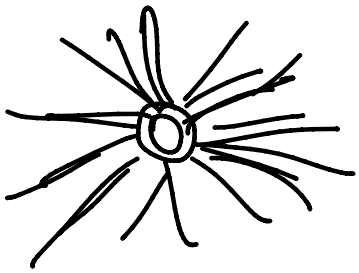


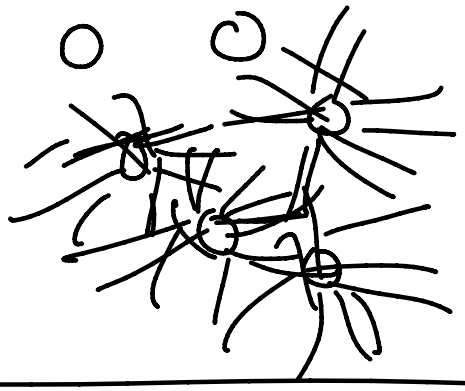
$|i_1 - i_2| + |j_1 - j_2|$

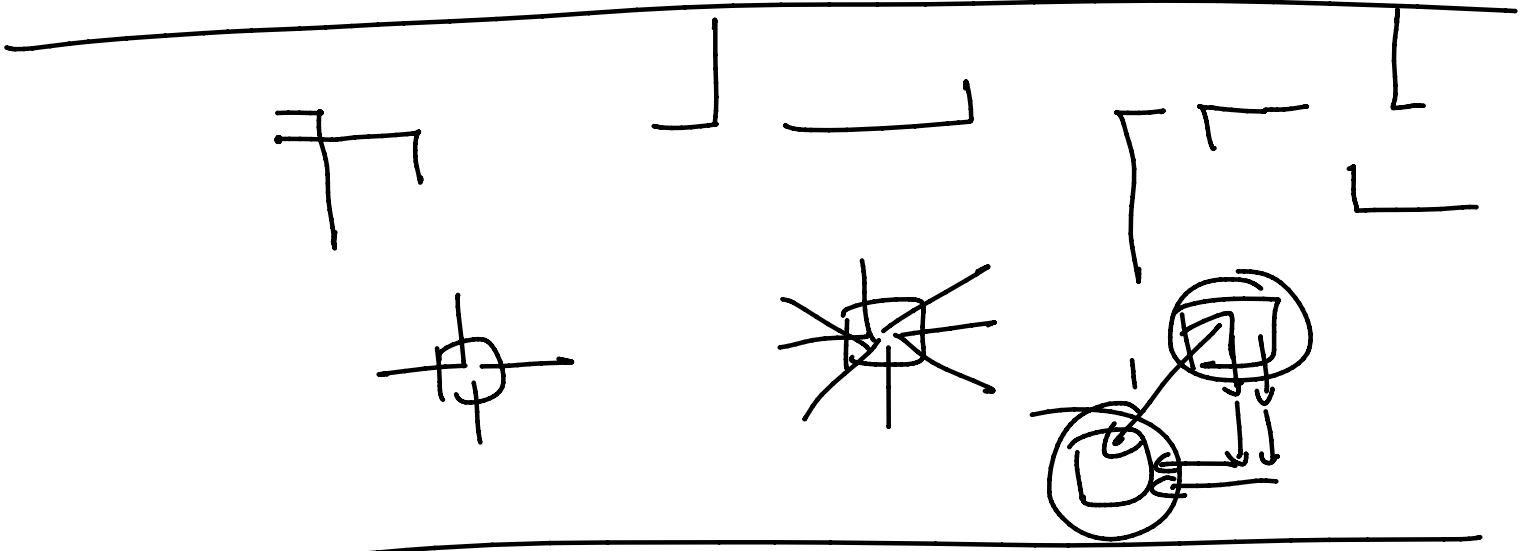


$n-1+m-1$

$n+m-2$





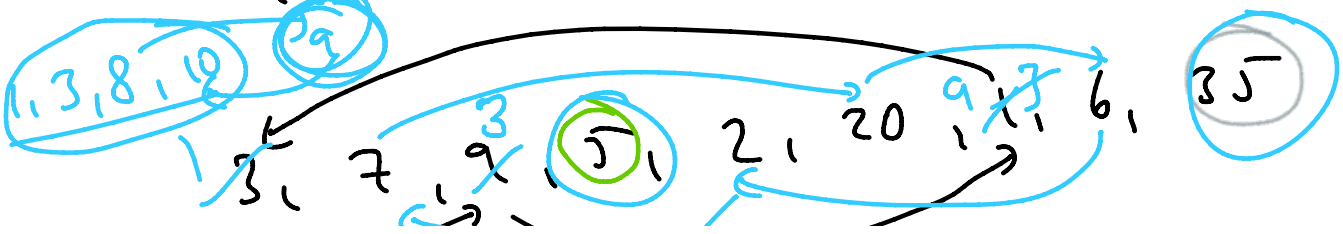


Q/. Minimum Swaps :-

$n = 9$

→ 3, 7, 9, 5, 2, 20, 1, 6, 35

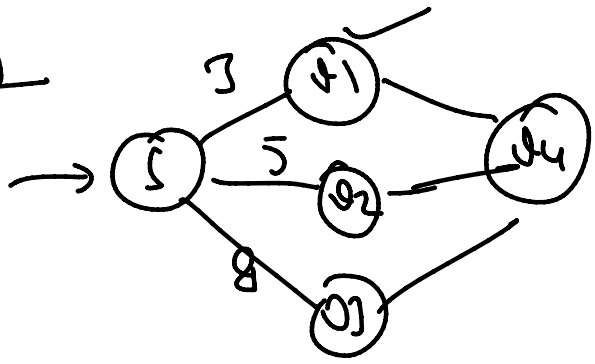
→ 1, 2, 3, 5, 6, 7, 9, 20, 35



$$\sum_{i=1}^{K} (\text{cycle-size } i-1)$$

⇒ Dijkstra Algo → to find shortest distance from source to all other vertices.

Greedy

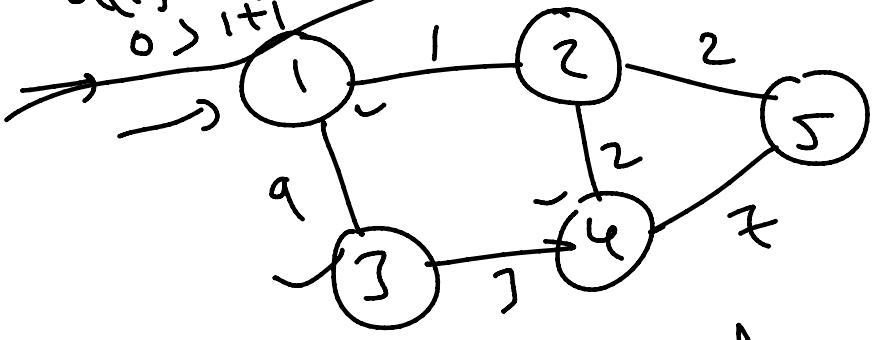


→ dis → ∞

dis [s] = 0 ;

d[1] > d(2)+1
0 > 1+1

d(2)+2
1+2



min priority q → val ↓ prio ↑
max → val ↑ prio ↑

⟨d, i⟩

	0	1	2	3	4	5
dis →	1	2	3	4	5	
	0	∞	∞	∞	∞	∞
	7	7	7	7	7	7
q →	~~(0,1)~~	~~(1,2)~~	(3,3)	~~(3,5)~~	(7,4)	(6,5)

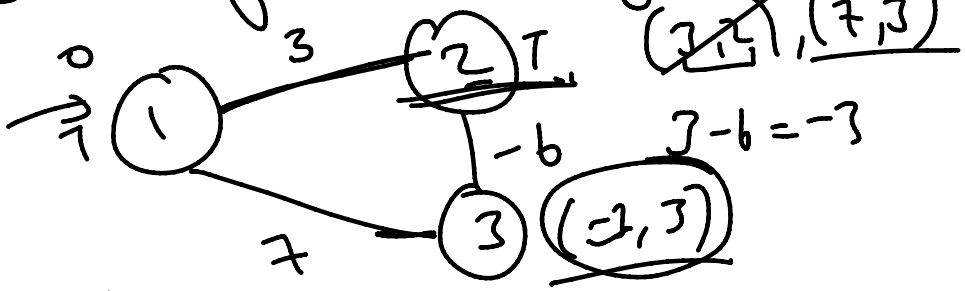
i=5 → 3
d=3 → 6
i=3 ✓
d=9

max ... → default pivot ...

< 5, 3, 8, 9 >

< 5, 3, 8, 9 (1) >

→ Not applicable for -ve edges.



⇒ $O((n+e)\log n)$