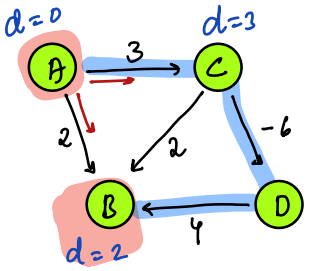
Todays Content:

- Bellman ford
- Floyd Warshal Algorithm

Dijkstra's With Negative Edges:

Ex:

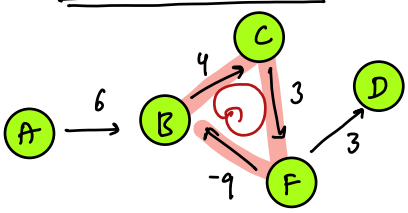


A → B :

{ According to dijkstras, length of Shortest path from A → B : 2

{ Correct ans: A $\xrightarrow{3}$ C $\xrightarrow{-6}$ D $\xrightarrow{4}$ B : 1

Negative Cycle weight



A → D : A → B → C → F → D : 16

: A → B → C → F → B → C → F → D : 14

-2

: Keep going in -ve cost decrease

: If -ve presents in graph, shortest path is not defined

Dijkstras Idea:

→ Blast the node with min value

→ Update all Adjacent Nodes

New Idea : { Bellman Ford } → +ve, -ve, We won't have -ve cycles

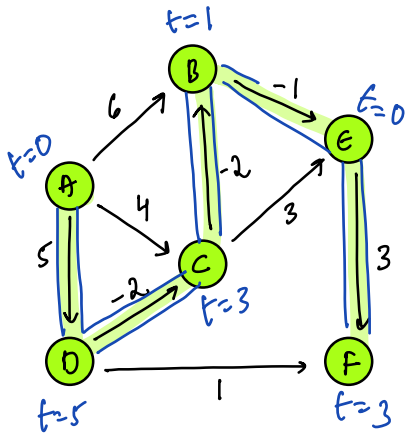
→ Iterate in every Edge & update nodes

→ Repeat above prom n-1 times

// n represents no: of nodes

TODO

Bellman ford : {no-ve cycle}



dist:	A	B	C	D	E	F
→	0	∞	∞	∞	∞	∞
itr1 :	0	2	3	5	5	6
itr2 :	0	1	3	5	1	4
itr3 :	0	1	3	5	0	3
itr4 :	0	1 ✓	3 ✓	5 ✓	0 ✓	3 ✓

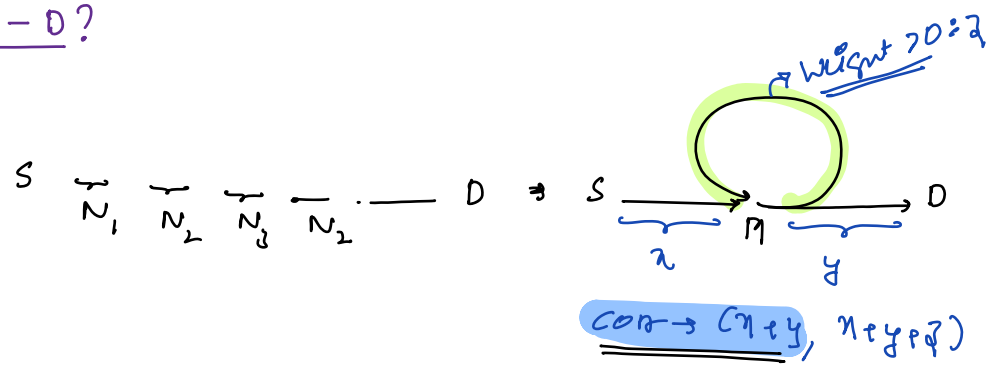
itr5 : [itr3 → itr4 no change we can break]

Edges:

$A \xrightarrow{6} B$, $A \xrightarrow{4} C$, $A \xrightarrow{5} D$, $B \xrightarrow{-1} E$, $C \xrightarrow{3} E$, $C \xrightarrow{-2} B$

$D \xrightarrow{-2} C$, $D \xrightarrow{1} F$, $E \xrightarrow{3} F$

N Nodes: At max what can be length of path from S − D? = N-1



cost → (x+y, x+y+2)

Pseudocode :

$u \xrightarrow{w} v$

```
Bellman ford ( list< pair< int, pair< int, int> > Edges, int N, int s){
    int dist[N+1] = {+∞};
    dist[s] = 0
    int e = Edges.length
    for(int k=1; k< N; k++){
        bool upd = false
        for(int i=0; i< e; i++){
            pair< int, pair< int, int> > data = Edges[i]
            int u = data.first
            int v = data.second.first
            int w = data.second.second
            if( dist[u]+w < dist[v] && dist[u] != +inf)
                dist[v] = dist[u]+w // change of overflow
                upd = true // indicates data changing
        }
        if( upd == false){
            break
        }
    }
}
```

$u \xrightarrow{w} v$

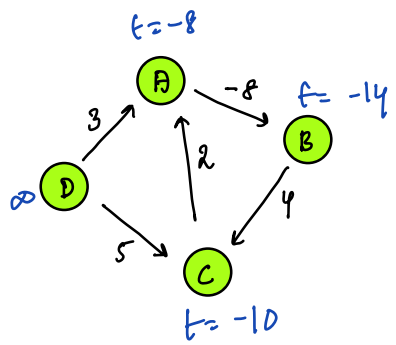
TC: $E \times N \Rightarrow O(N*E)$

SC: $O(N)$

-ve cycle detection Using Bellman Ford:

Q) Given directed graph, check if -ve cycle is present or not

(-ve)



dist:	A	B	C	D
→	0	∞	∞	∞
Itr1:	-2	-8	-4	∞
Itr2:	-4	-10	-6	∞
Itr3:	-6	-12	-8	∞
Itr4:	-8	-14	-10	0

: It should not change

obs: At 4th iteration data still changes, 100% -ve cycle present

Edges:

A $\xrightarrow{-8}$ B B $\xrightarrow{4}$ C C $\xrightarrow{2}$ A D $\xrightarrow{3}$ A D $\xrightarrow{5}$ C

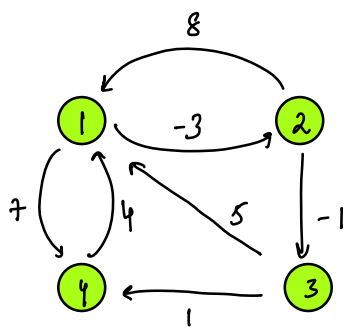
In general: If dist[] is changing at N^{th} iteration as well in that case we can say that, there is -ve weight cycle

TC: $N*E$, SC: $O(N)$

Floyd Warshall {All pair shortest path} : [Between every 2 nodes find shortest path]

↳ dp

↳ Graph can contain -ve Edges but not -ve cycles



Idea: Run bellman ford from every node

↳ TC: $N * [N * E] \Rightarrow O(N^2 E)$

$dp[i, j, k]$ = { Min cost from $i \to j$, such node in between can only be $[1, 2, \ldots k]$ }

Ex: $dp[i\ j\ 3]$ = Min cost from $i \to j$, $\boxed{\text{node in between } [1, 2, 3]}$

↳ $dp[i, j, k]$ = i $\underbrace{[1\ 2\ 3 \ldots k]}$ j

k not present: $i\ \{1\ 2 \ldots k-1\}\ j$

k present: $i\ \{1, 2, \ldots k-1\}\ k\ \{1, 2 \ldots k-1\}\ j$

$$dp[i, j, k] = \min\left[dp[i, j, k-1] \quad dp[i, k, k-1] + dp[k, j, k-1]\right]$$

$k \to k-1 \to k-2 \to k-3 \to \ldots$

Pseudocode:

```
int dp[N+1][N+1][2];
k=1; k<=n; k++){
    i=1; i<=n; i++){
        j=1; j<=n; j++){
            dp[i, j, k%2] = min (dp[i, j, (k-1)%2)
                                 dp[i, k, (k-1)%2] + dp[k, j, (k-1)%2]
        }
    }
}
```

Base Condition $k=0$ = { Adj Matrix }

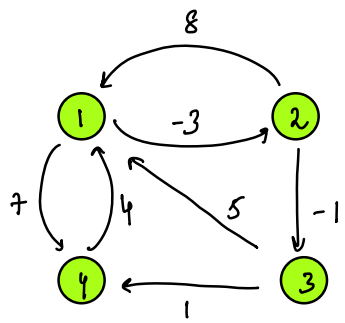
$dp[i, j, 0]$ = min cost to reach from $i-j$, in between no nodes

We can optimize to $O(N^2)$

TC: $O(N^3)$ SC: $O(N^3) \rightarrow O(N^2)$

dp[5][5][5]

k=0: [Base case]



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

k=1: → [i, j]: min cost to i-j, between {1}

	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	0	-3	∞	7
2	∞	8	0	-1	15
3	∞	5	2	0	1
4	∞	4	1	∞	0

k=2: → [i, j]: min cost to i-j, between {1, 2}

	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	0	-3	-4	7
2	∞	8	0	-1	15
3	∞	5	2	0	1
4	∞	4	1	0	0

value can be wrong
please cross verify once

$k=3$: → [i, j]: min cost to i−j, between {1, 2, 3}

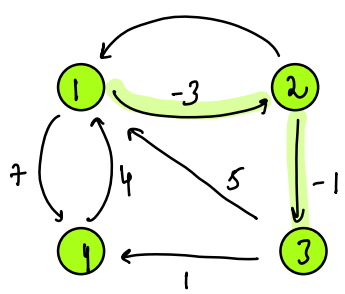
	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	0	−3 ✓	−4	−3
2	∞	4 ✓	0	−1	0 ✓
3	∞	5	2	0	1
4	∞	4 ✓	1 ✓	0	0

value can be wrong please cross verify once

$k=4$: → [i, j]: min cost to i−j, between {1, 2, 3, 4}

	0	1	2	3	4
0	∞	∞	∞	∞	∞
1	∞	0	−3	−4	−3
2	∞	4	0	−1	0
3	∞	5	2	0	1
4	∞	4	1	0	0

value can be wrong please cross verify once



- → { of Strongly Connected Components }
- → { Articulation points & Edges }
- → Euler paths
- → Segment Tree { Tree, Recursion }

→ Company:
↳ [previous asked questions on leetcode]

- → Revise notes
  - ↳ Problems we discussed / Assignment / Homework
  - ↳ Previous asked question in company