Bellman Ford and Floyd Warshall

14 August 2023 21:58

Today :-

- Bellman Ford
- Floyd Warshall

Recall :-

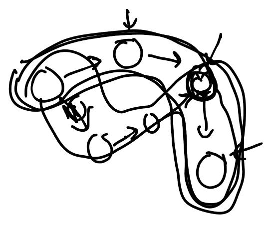
- Dijkstra → $\underbrace{\text{single short shortest path}}$

Dijkstra → infinite loop if there is any negative edge then it would not work

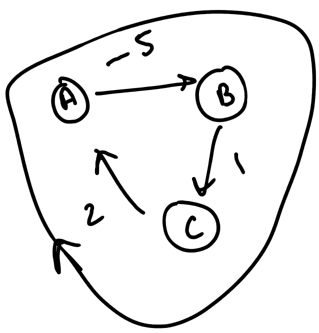
$\boxed{TC \rightarrow O(V + E \log E)}$ → excellent

Bellman Ford → does the same work negatives edges ✓

Bellman Ford → negative cycles or not



- → more power~~r~~ful
- → but ~~more~~ ~~costly~~ costlier



negative weight

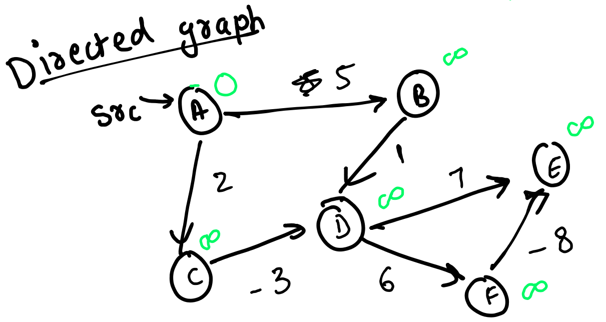
$-\infty$

$\sum wt < 0$

negative cycle

Bellman Ford

Directed graph

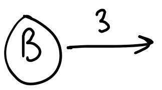


u→v

edges

dist[v] + wt

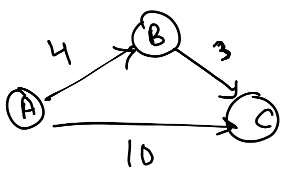
B → dist[B] → 4



~~dist[C]~~ 4 + 3 = 7

< dist[C]

$\boxed{dist[C] = dist[B] + wt}$



dist[C]=10

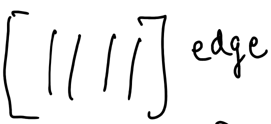
dist[B]=4

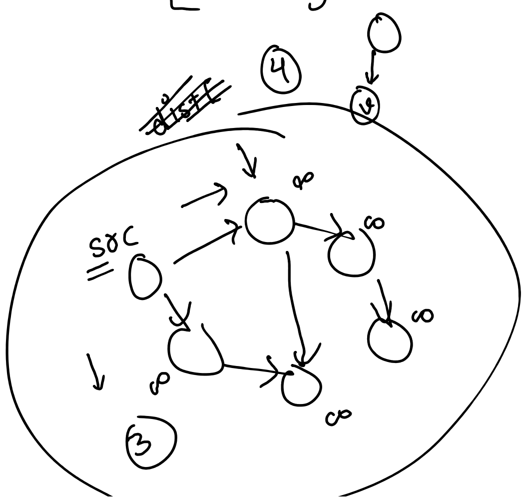
dist

$B \xrightarrow{wt} C$

dist[B] + wt < dist[C]

dist[node] → the minimum weight you can have to go to node from src





$u \xrightarrow{wt} v$

dist[v] = min(dist[u] + wt, dist[v]);

atleast one node which go to the most optimal distance

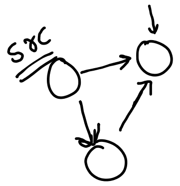
③ ∞

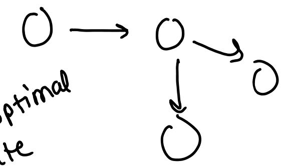
Principle ~~of~~ of Mathematical Induction (PMI)

① src → v

One node will be getting to most optimal dist

it is not getting to the most optimal state





ans ~~D~~ dist[u] → min. dist. to node u from src

then it means it would n be getting from some other path

dist[u] → dist[v] →

got some idea

Algorithm

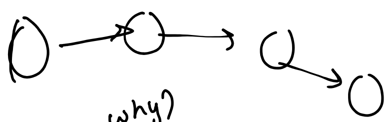
relaxation of a vertex

→ dist[v] > dist[u] + wt

dist[v] =

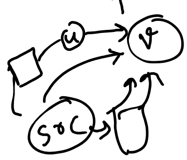
(n-1)

~~rela~~



why?

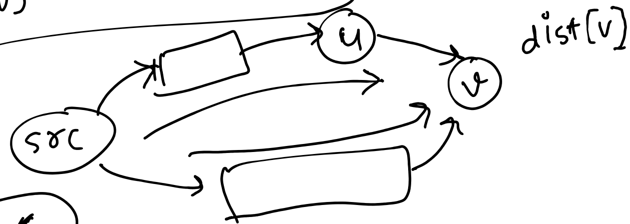
```
for (int i=0; i<n-1; i++){
    for (auto [u, v, wt] : edges){
        if (dist[u] != INF){
            dist[v] = min(dist[v], dist[u]+wt);
        }
    }
}
```



dist[v] = min(   );

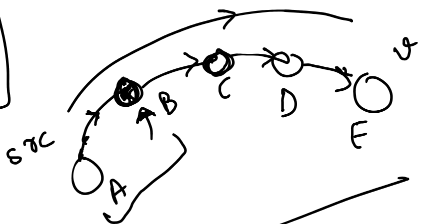
for all the vertices which are having their shortest paths with length $\leq l$

$\boxed{dist[v] = min(\quad);}$



$\boxed{dist[node]}$ ✓

$\boxed{1}$ [A → B]



5

from A to B

✓ A → B

$\boxed{\text{src } A \longrightarrow B}$

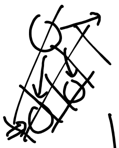
dist[B] = dist[A] + wt

1 node for every iteration → $n-1$ iterations

1st → B
2nd → C
3rd → D
4th → E

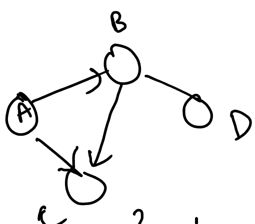
A → B
B → C
C → D
D → E

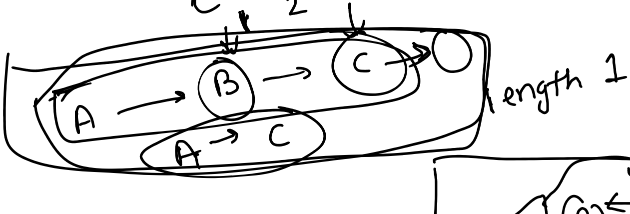
B → C



edges

A → B
A → C





n

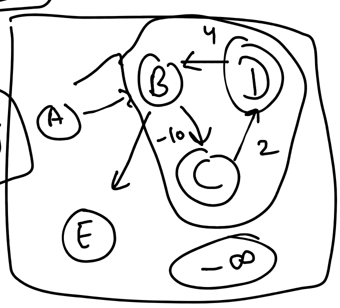
$n-1$

length n → cycle

negative cycle

∞ length

∞ length



for any node

if the shortest path from source is of length $\leq l$

l^{th} iteration

it would definitely have the shortest path achieved

dist

path with $\leq n-1$

$n-1$

n^{th} iteration

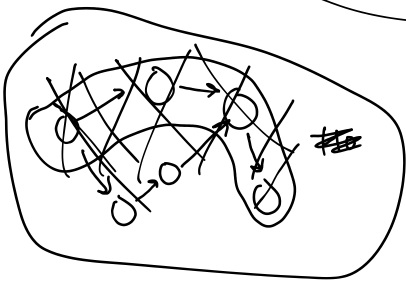
if for ~~any~~ ~~any~~ any edge

~~nth~~ path $\geq n$ size

negative cycle must be there

$u \xrightarrow{wt} v$

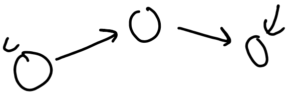
$dist[v] > dist(u) + wt$



negative edges are not causing any trouble

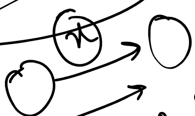
Think about it

PMI



src

pos & neg



same edge multiple

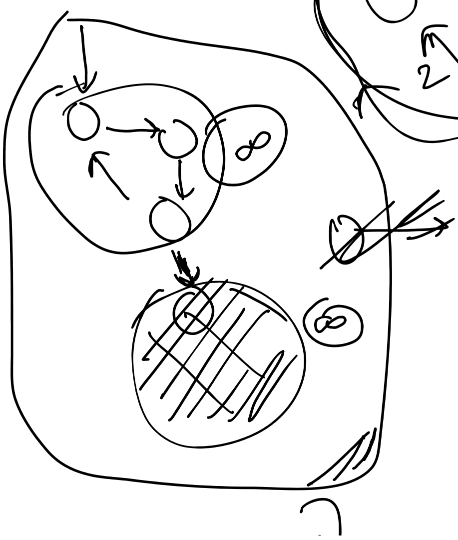
relaxation

from[v] → u

∞

Bellman ford

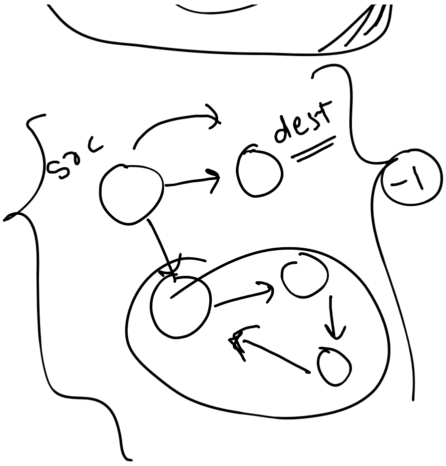
positive cycle b/w src & dest



all the nodes which are a part of negative cycles

$O(V * E) \rightarrow$ $V \leq 1000$

$E \leq 10^5$



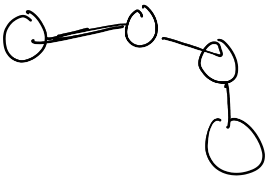
$O(V * E) \rightarrow$ $V \le 1000$, $E \le 10^5$

idea

understood Bellman Ford

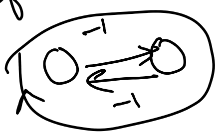
directed edges ?
undirected edges

m n*m V*E
E



positive → dijkstra

if there is any negative



[Problem]()

$\boxed{V^3}$ ×

Floyd Warshall

$O(n^3) \rightarrow n \le 100$

short path from every node to every node

multi source shortest path

bellman ford

for every node apply bellman ford

$V * (V * E)$ $\quad E \le \frac{V*(V-1)}{2}$

$V^2 * E$ $\quad V^2$

$O(V^4)$

dijkstra

$V * (V + E \log E)$

$V^2 + VE \log E \rightarrow V^2 \log V$ $\quad E \approx V^2$

neg edges

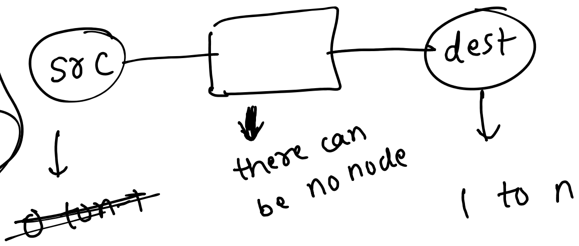
Floyd Warshall (DP)

$dist[u][v]$

$dist[i][i] = 0$

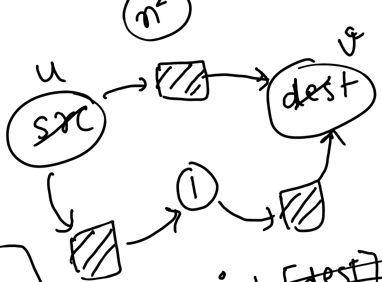
$u \xrightarrow{wt} v$

$dist[u][v] = wt$



~~DP~~ ~~base~~

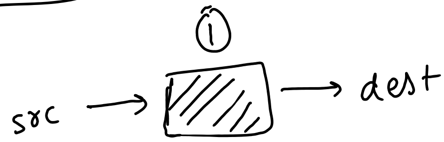
n^2



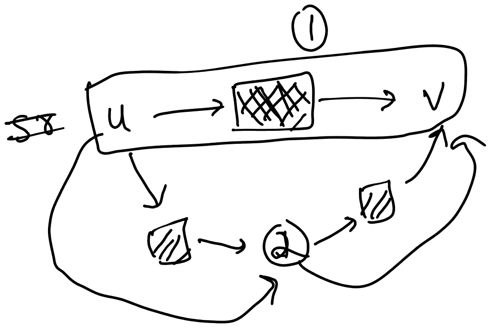
$dist[u][1] + dist[1][v]$

~~dist[dest]~~

$dist[u][v] \rightarrow$ ~~short~~ smaller

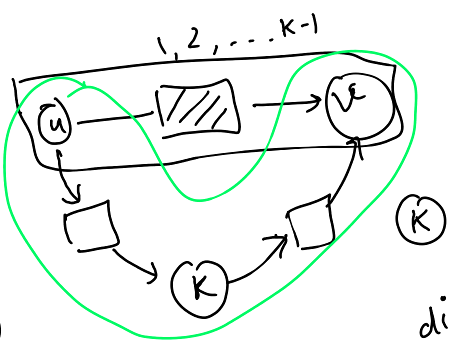


$dist[u][v]$



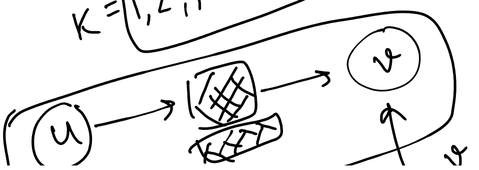
$dist[u][v] = \min(dist[u][v],\ dist[u][2] + dist[2][v]);$

3, 4, ..., K

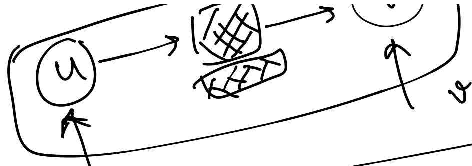


$dist[u][v] = \min(dist[u][v],\ dist[u][k] + dist[k][v]);$

K = 1, 2, ,. . . n



$dist[u][v] \rightarrow dist[k][u][v]$



u v

dist[u][v] →

dist[k][u][v]

~~dist[k][v][u]~~

negative weight edges →

brute forcing

dist[k][u][v] → u to v   where first k nodes can be visited in the path

dist[u][v]

1 to n

0 to n-1

```
vvi dist(n, vi(m, INF));   // INF written over crossed-out 1e9
for (int i = 0; i < n; i++){
    dist[i][i] = 0;
}

for all edge [u, v, wt] :-
    dist[u][v] = wt;
    ~~dist[v][u] = wt;~~

for ( int k=0; k<n; k++){
    for( int i=0 ; i<n; i++){
        for (int j = 0; j<n; j++) {
            if (dist[i][k] != INF && dist[k][j] != INF)
            { dist[i][j] = min( dist[i][j],
                          dist[i][k] + dist[k][j]);
            }
        }
    }
}
```

what if there is a negative cycle

INF = 1e9

$u \xrightarrow{wt} v$

wt → < 0

dist

What

i

$dist[i][i]$ = less than 0

$u \rightarrow v$

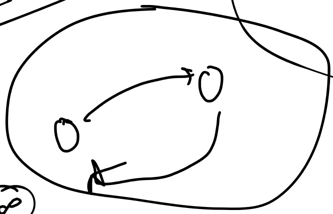
dist

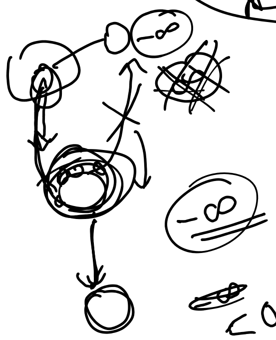
wt → INF

$u \xrightarrow{-1} K \xrightarrow{INF} v$

INF

INF - 1

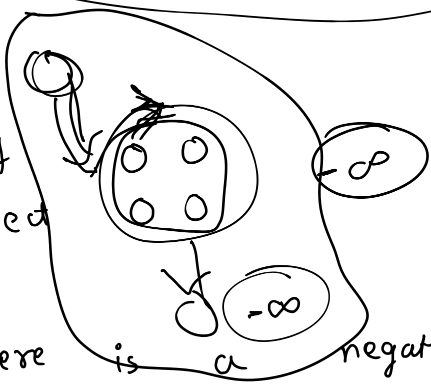




-∞

< 0

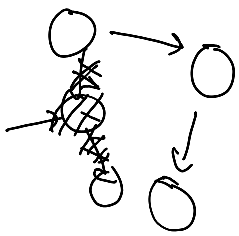
dist array is incorrect



-∞

-∞

if there is a negative cycle then Floyd Warshall also fails to find the short



graph

Luv

Luv CP

TLE

sum of shortest paths from all nodes to node

n

m

n

queries

Floydwarshall

$n^3 \rightarrow n^2$

bad

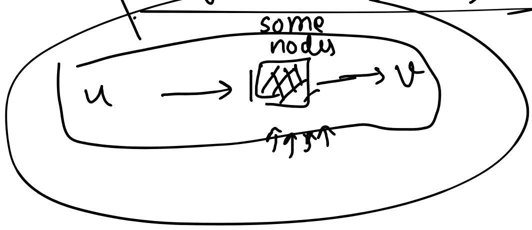
n^4

codeforces

we are removing nodes one by one

adding nodes one by one

floyd warshall



Problem