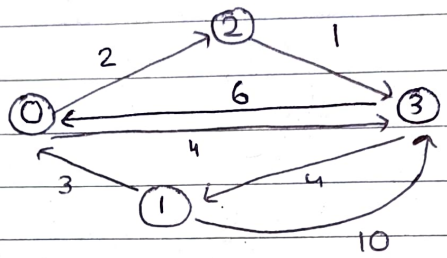
Date _____ Page _____

Day-211

Graph-15

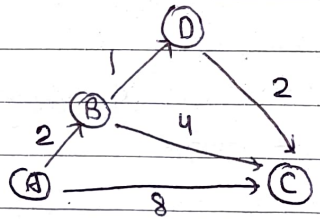
* **Floyd Warshall Algorithm:**



⇒ This algorithm is used for finding multiple source shortest path.

⇒ It is only applied to directed graph.

⇒ Let is take an example,



⇒ Suppose, we want to go to C from A, then, $A \xrightarrow{8} C$

But, $[A \xrightarrow{2} B \ , \ B \xrightarrow{4} C] = A \xrightarrow{6} C$

⇒ We can also take the above path.

This is logic behind FWA

Date _____ Page _____

⇒ FWA is based on the property that if there is a path between A & B and B & C.

⇒ Then, there must be path from A to C by going through B.

⇒ So, this thing or property is called Transitivity.

⇒ It is given by Warshall.

⇒ And the Floyd main task is to update the distance by choosing min distance.

⇒ So, for A → D,

A $\xrightarrow{2}$ B , B $\xrightarrow{1}$ D

A $\xrightarrow{3}$ D

⇒ This algo works on a very important principle that →

⇒ Small milate jao, large banate jao 😊 😀 (Rohit bhaiya rocks 🅾 🅾)

⇒ So, here comes a concept of intermediate nodes.

⇒ We will consider every other node as intermediate node and check for the

0
1
2
3

Date _____ Page _____

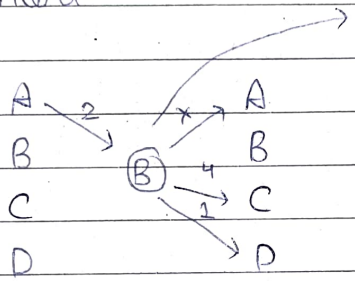
target node by inter. node.

⇒ Ex ⇒ : A → C

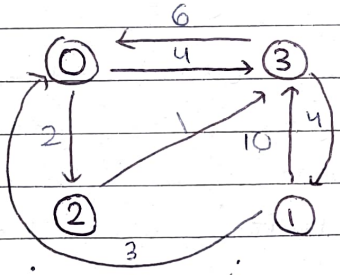
A $\xrightarrow{2}$ B , B $\xrightarrow{4}$ C (B as IN)

A $\xrightarrow{3}$ B, B $\xrightarrow{1}$ D, D $\xrightarrow{2}$ C (D as IN)

⇒ So, in short —

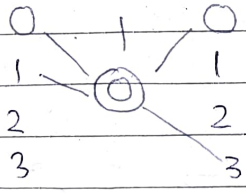


⇒ Ex:



	0	1	2	3
0	0	∞	2	4
1	3	0	∞	10
2	∞	∞	0	1
3	6	4	∞	0

⇒ So,



↓0

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

⇒ path(i,j) = min(path(i,j) ,
path(i,k) + path(k,i)); ↓1

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

3 ↓

	0	1	2	3
0	0	∞	2	3
1	3	0	5	6
2	∞	∞	0	1
3	6	4	8	0

2 ↓

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

Date _____ Page _____

⇒ In FWA, ordering doesn't matter of taking intermediate node.

⇒ Suppose, we are taking '2' as a ~~In~~ intermediate node between i & j then it is also possible that other intermediate nodes are present.

Code

```
for( int k=0; k<n; k++){
    for( i=0; i<n; i++){
        for( j=0; j<n; j++){
            mat[i][j] = min( mat[i][j],
                mat[i][k] + mat[k][j] )
        }
    }
}
```

Time Complexity → $O(V^3)$
Space Complexity → $O(1)$

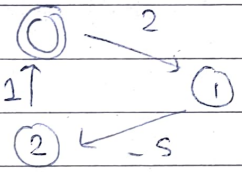
⇒ FWA can also detect negative cycle and can also work for -ve weights.

Date _____ Page _____

⇒ So, if the diagonal element that we always have zero value but have any -ve value.

⇒ That means -ve cycle present in the graph.

Ex:



⇒ $1 \overset{0}{\rightarrow} 1$ (in (but by 2 as IN)

⇒ $1 \underset{-5}{\rightarrow} 2 \underset{3}{\rightarrow} 1 \rightarrow -2$

⇒ $1 \overset{-2}{\rightarrow} 1$ (but self distance can't be -ve)

⇒ So, -ve cycle present.

Dijkstra: Only for +ve

	Sparse	Dense
⇒	$O(E \log V) * V$	$O(V^2) * V$
⇒	$O(VE \log V)$	$O(V^3)$
⇒	$V^2 \log V$	$O(V^3)$

⇒ FW: Easy to implement.