Floyd-Warshall:

The Floyd-Warshall Method is a weighted graph algorithm that finds the shortest path between all pairs of vertices.

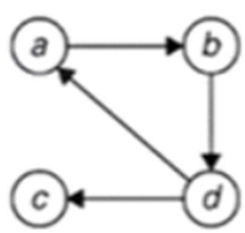
Transitive Closure:

This algorithm works for weighted graphs that are both directed and undirected.It does not, however, operate for graphs with negative cycles (where the sum of the edges in a cycle is negative).

The reachability matrix to go from vertex u to vertex v of a graph is known as transitive closure. We are given a graph and must find a vertex v that can be reached from another vertex u for all vertex pairs (u, v).

Transitive matrix is n X n matrix where if value of 1st row and 1st column is 1 then from 1st node then we can reach 1st node likewise if 1st row and 3rd column is 1 then we can say that we can reach 3rd node from the 1st node or if value is 0 then it means we can't reach Cth node from Rth column. here 1st node is A and 3rd node is C.

Ex.



We have a graph here, and we will develop a transitive closure for it. To find Closure, we must first determine the reachability of all its nodes.

For A:

The first row of transitive closure is A's reachability.

A: Obviously, we can go to A from itself, therefore we'll put 1 in the first column of the first row.

B: As seen in the graph, we may access B node from A, hence we will enter value 1 for the second column of the first row.

C: To go from A to C, we must traverse the graph in the following order: A->B->D->C. Following this method, we can also go to C from A, so we enter 1 for the third column of the first row.

D: We can also go to D by taking the A->B->D path. As a result, we will also enter 1 for the fourth column of the first row.

So our 1st row of matrix will look like **1 1 1 1**

For B:

The second row of transitive closure is B's reachability.

A: As seen in the graph, we can go to A node from B by taking the B->D->A path. As a result, we will enter 1 for the first column of the second row.

B: Obviously, we can go to B from itself, therefore we'll put 1 in the second column of the second row.

C: Now, in order to go from B to C, we must traverse the graph in the following order: B->D->C. Following this approach, we can also go to C from B, so we write 1 in the third column of the second row.

D: We can also get to D by taking the B->D route. As a result, we will also enter 1 for the fourth column of the second row.

So our 2nd row of matrix will look like **1 1 1 1**

For C:

C's reachability is the third row of the Transitive closure.

A: As seen in the graph, we cannot access A node from C. As a result, we will enter 0 in the first column of the third row.

B: As seen in the graph, we cannot access B node from C. As a result, we will enter a value of 0 for the second column of the third row.

C: Obviously, we can go to C from itself, therefore we'll put 1 in the third column of the third row.

D: As seen in the graph, we cannot access B node from C. As a result, we will enter a value of 0 for the fourth column of the third row.

So our 3nd row of matrix will look like **0 0 1 0**

For D:

D's reachability is the fourth row of the Transitive closure.

A: As seen in the graph, we can go to A node from D by taking the D->A path. As a result, we shall enter 1 for the first column of the fourth row.

B: We can also go to B by using the D->A->B route. As a result, we'll enter 1 for the second column of the fourth row as well.

C: We can also get to C by taking the D->C approach. As a result, we will also enter 1 for the third column of the fourth row.

D: Obviously, we can go to D from itself, therefore we'll place 1 in the 4th column of the 4th row.

So our 4th row of matrix will look like **1 1 1 1**

So, according to our estimates, the ultimate transitive closure will be

1 1 1 1
1 1 1 1
0 0 1 0
1 1 1 1

ps: we don't have to verify the entire path from point 1 to point 2 to see whether we can get to point 2 from point 1.

For example, if we want to find out if we can get to B from D, we don't have to verify the entire path; we simply need to see if we can get to A from D because it is already known that we can go to B from A.

Floyd-Warshall:

The Floyd-Warshall Method is a weighted graph algorithm that finds the shortest path between all pairs of vertices.

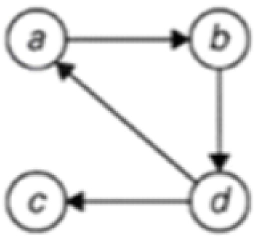
Transitive Closure:

This algorithm works for weighted graphs that are both directed and undirected.It does not, however, operate for graphs with negative cycles (where the sum of the edges in a cycle is negative).

The reachability matrix to go from vertex u to vertex v of a graph is known as transitive closure. We are given a graph and must find a vertex v that can be reached from another vertex u for all vertex pairs (u, v).

Transitive matrix is n X n matrix where if value of 1st row and 1st column is 1 then from 1st node then we can reach 1st node likewise if 1st row and 3rd column is 1 then we can say that we can reach 3rd node from the 1st node or if value is 0 then it means we can't reach Cth node from Rth column. here 1st node is A and 3rd node is C.

Ex.



We have a graph here, and we will develop a transitive closure for it. To find Closure, we must first determine the reachability of all its nodes.

For A:

The first row of transitive closure is A's reachability.

A: Obviously, we can go to A from itself, therefore we'll put 1 in the first column of the first row.

B: As seen in the graph, we may access B node from A, hence we will enter value 1 for the second column of the first row.

C: To go from A to C, we must traverse the graph in the following order: A->B->D->C. Following this method, we can also go to C from A, so we enter 1 for the third column of the first row.

D: We can also go to D by taking the A->B->D path. As a result, we will also enter 1 for the fourth column of the first row.

So our 1st row of matrix will look like **1 1 1 1**

For B:

The second row of transitive closure is B's reachability.

A: As seen in the graph, we can go to A node from B by taking the B->D->A path. As a result, we will enter 1 for the first column of the second row.

B: Obviously, we can go to B from itself, therefore we'll put 1 in the second column of the second row.

C: Now, in order to go from B to C, we must traverse the graph in the following order: B->D->C. Following this approach, we can also go to C from B, so we write 1 in the third column of the second row.

D: We can also get to D by taking the B->D route. As a result, we will also enter 1 for the fourth column of the second row.

So our 2^{nd} row of matrix will look like **1 1 1 1**

For C:

C's reachability is the third row of the Transitive closure.

A: As seen in the graph, we cannot access A node from C. As a result, we will enter 0 in the first column of the third row.

B: As seen in the graph, we cannot access B node from C. As a result, we will enter a value of 0 for the second column of the third row.

C: Obviously, we can go to C from itself, therefore we'll put 1 in the third column of the third row.

D: As seen in the graph, we cannot access B node from C. As a result, we will enter a value of 0 for the fourth column of the third row.

So our 3^{nd} row of matrix will look like **0 0 1 0**

For D:

D's reachability is the fourth row of the Transitive closure.

A: As seen in the graph, we can go to A node from D by taking the D->A path. As a result, we shall enter 1 for the first column of the fourth row.

B: We can also go to B by using the D->A->B route. As a result, we'll enter 1 for the second column of the fourth row as well.

C: We can also get to C by taking the D->C approach. As a result, we will also enter 1 for the third column of the fourth row.

D: Obviously, we can go to D from itself, therefore we'll place 1 in the 4th column of the 4th row.

So our 4^{th} row of matrix will look like **1 1 1 1**

So, according to our estimates, the ultimate transitive closure will be

1 1 1 1
1 1 1 1
0 0 1 0
1 1 1 1

ps: we don't have to verify the entire path from point 1 to point 2 to see whether we can get to point 2 from point 1.

For example, if we want to find out if we can get to B from D, we don't have to verify the entire path; we simply need to see if we can get to A from D because it is already known that we can go to B from A.