Floyd-Warshall Algorithm

Let the vertices of G be V = {1, 2........n} and consider a subset {1, 2........k} of vertices for some k. For any pair of vertices i, j ∈ V, considered all paths from i to j whose intermediate vertices are all drawn from {1, 2.......k}, and let p be a minimum weight path from amongst them. The Floyd-Warshall algorithm exploits a link between path p and shortest paths from i to j with all intermediate vertices in the set {1, 2.......k-1}. The link depends on whether or not k is an intermediate vertex of path p.

If k is not an intermediate vertex of path p, then all intermediate vertices of path p are in the set {1, 2........k-1}. Thus, the shortest path from vertex i to vertex j with all intermediate vertices in the set {1, 2.......k-1} is also the shortest path i to j with all intermediate vertices in the set {1, 2.......k}.

If k is an intermediate vertex of path p, then we break p down into i → k → j.

Let dij(k) be the weight of the shortest path from vertex i to vertex j with all intermediate vertices in the set {1, 2.......k}.

A recursive definition is given by

Floyd-Warshall Algorithm

**FLOYD - WARSHALL (W)**

n ← rows [W].

D0 ← W

for k ← 1 to n

do for i ← 1 to n

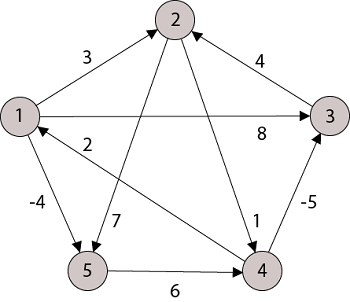
do for j ← 1 to n

do dij(k) ← min (dij(k-1),dik(k-1)+dkj(k-1) )

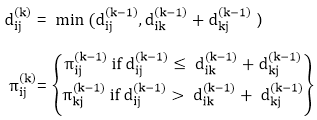
return D(n)

The strategy adopted by the Floyd-Warshall algorithm is **Dynamic Programming**. The running time of the Floyd-Warshall algorithm is determined by the triply nested for loops of lines 3-6. Each execution of line 6 takes O (1) time. The algorithm thus runs in time θ(n3 ).

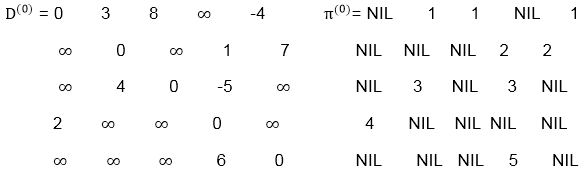
**Example:** Apply Floyd-Warshall algorithm for constructing the shortest path. Show that matrices D(k) and π(k) computed by the Floyd-Warshall algorithm for the graph.



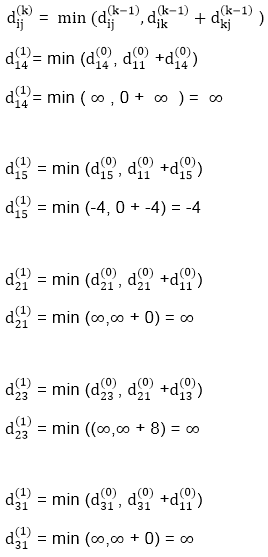
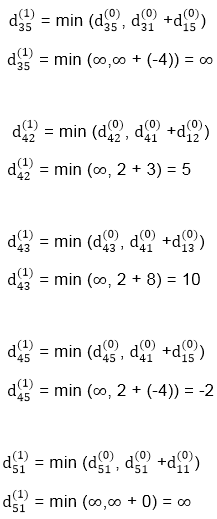
**Solution:**

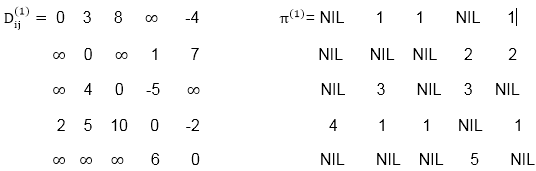


**Step (i)** When k = 0

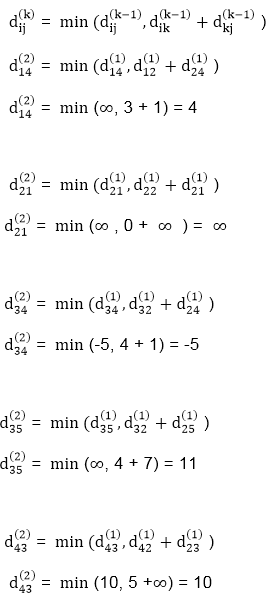


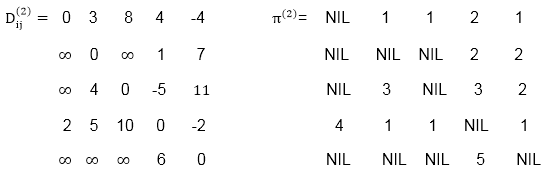
**Step (ii)** When k =1

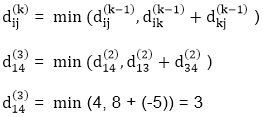


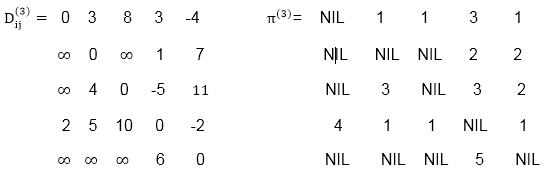
**Step (iii)** When k = 2



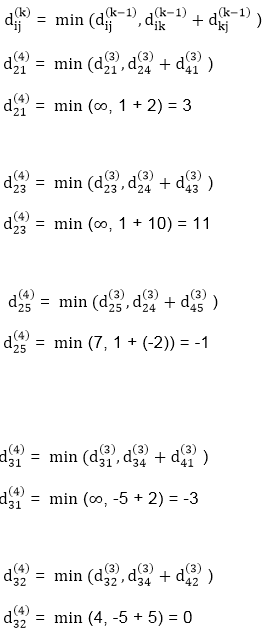
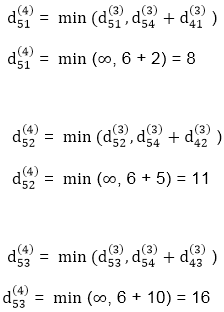


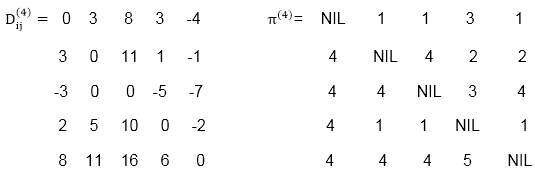
**Step (iv)** When k = 3



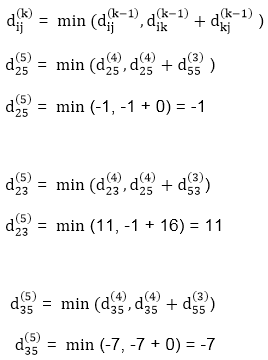


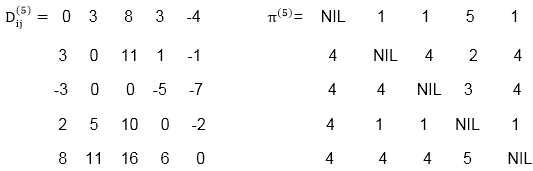
**Step (v)** When k = 4

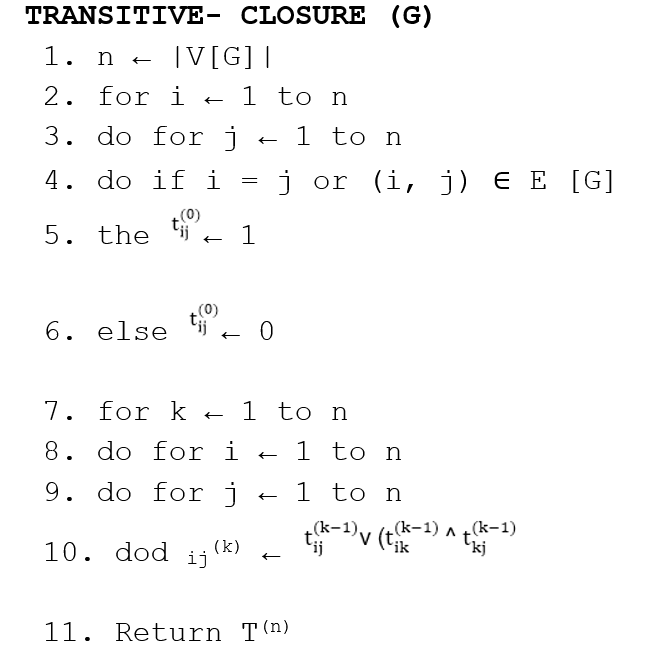
  




**Step (vi)** When k = 5





****

**RELEVANT READING MATERIAL AND REFERENCES:**

**Source Notes:**

1. https://www.javatpoint.com/floyd-warshall-algorithm

**Lecture Video:**

1. <https://youtu.be/oNI0rf2P9gE>

**Online Notes:**

1. <http://vssut.ac.in/lecture_notes/lecture1428551222.pdf>

**Text Book Reading:**

1. Cormen, Leiserson, Rivest, Stein, “*Introduction to Algorithms*”, Prentice Hall of India, 3rd edition 2012. problem, Graph coloring.

**In addition: PPT can be also be given.**