All Pair Shortest path

Also known as Floyd-Warshall algorithm

All Pair Shortest Path

- The all-pairs shortest path problem is the determination of the shortest graph distances between every pair of vertices in a given graph. The problem can be solved using applications of Dijkstra's algorithm or all at once using the Floyd-Warshall algorithm. The latter algorithm also works in the case of a weighted graph where the edges have negative weights.
- The matrix of all distances between pairs of vertices is called the graph distance matrix, or sometimes the all-pairs shortest path matrix.

All Pair Shortest Path

- Use a different dynamic-programming formulation to solve the all-pairs shortest-paths problem on a directed graph G=(V,E).
- The resulting algorithm, known as the Floyd-Warshall algorithm, runs in O (V^3) time.
  - negative-weight edges may be present,
  - but we shall assume that there are no negative-weight cycles.
- The algorithm considers the “intermediate” vertices of a shortest path, where an intermediate vertex of a simple path p=<v1,v2,…,vl> is any vertex in p other than v1 or vl, that is, any vertex in the set {v2,v3,…,vl-1}

All Pair Shortest Path

- Let G=(V,E) be a weighted directed graph with n vertices.
- Let cost be a cost adjacency matrix for G such that cost(i,i)=0, for 1<=i<=n.
  - cost(i,j) is the length(or cost) of the edge <i,j>, if <i,j> ∈ E(G) and
  - cost = infinity if i ≠ j and <i,j> ∉ E(G).
- All pair shortest path is to find shortest paths between all paths of vertices in a graph i.e., to determine a matrix A such that A(i,j) is the length of a shortest path from i to j.

Input and Output format

- Input:
  - We assume V={1,2,3……n}
  - Assume that the graph is represented by an n x n matrix with the weights of the edges:

$$w_{ij} = \begin{cases} 0 & \text{if } i = j, \\ w(i,j) & \text{if } i \neq j \text{ and } (i,j) \in E, \\ \infty & \text{if } i \neq j \text{ and } (i,j) \notin E. \end{cases}$$

- Output: an n x n matrix $D = [d_{ij}]$ where d_{ij} is the length of the shortest path vertex i to j.

- 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, consider 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 among them.
- The Floyd-Warshall algorithm exploits a relationship between path p and shortest paths from i to j with all intermediate vertices in the set {1,2,…,k-1}.

- The relationship 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, a shortest path from vertex i to vertex j with all intermediate vertices in the set {1,2,…,k-1} is also a shortest path from 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 $\xrightarrow{p1}$ k $\xrightarrow{p2}$ j as shown Figure 2.p1 is a shortest path from i to k with all intermediate vertices in the set {1,2,…,k-1}, so as p2.

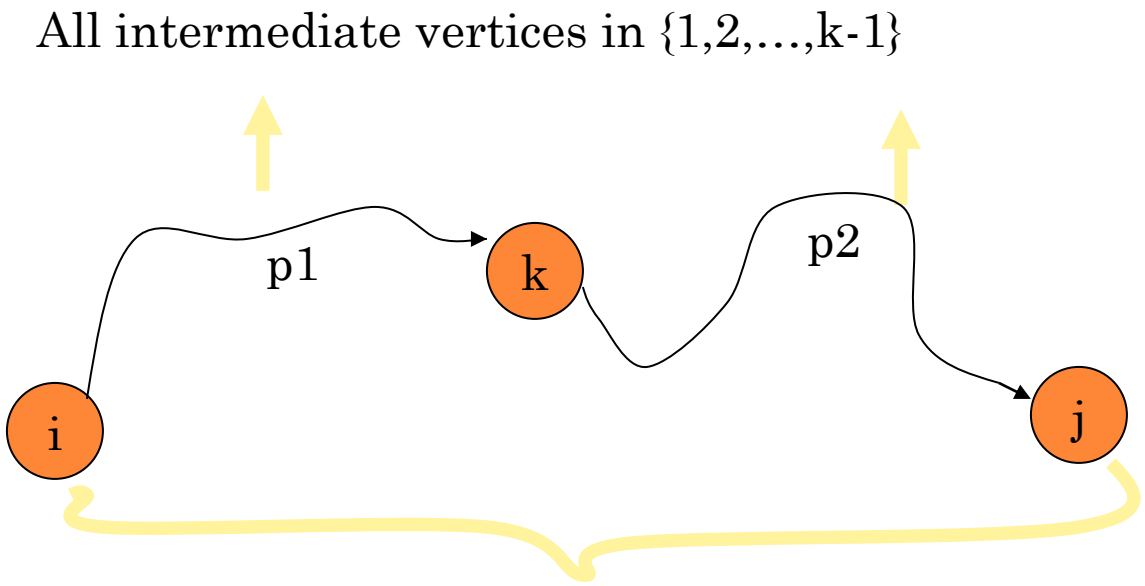


Figure shows: Path p is a shortest path from vertex i to vertex j,and k is the highest-numbered intermediate vertex of p. Path p1, the portion of path p from vertex i to vertex k,has all intermediate vertices in the set {1,2,…,k-1}.The same holds for path p2 from vertex k to vertex j.

A RECURSIVE SOLUTION TO THE ALL-PAIRS SHORTEST PATHS PROBLEM:

- Let $d_{ij}^{(k)}$ be the weight of a shortest path from vertex i to vertex j with all intermediate vertices in the set {1,2,…,k}. A recursive definition is given by
- $$d_{ij}^{(k)} = \begin{cases} w_{ij} & \text{if } k=0, \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{if } k \quad 1. \end{cases}$$

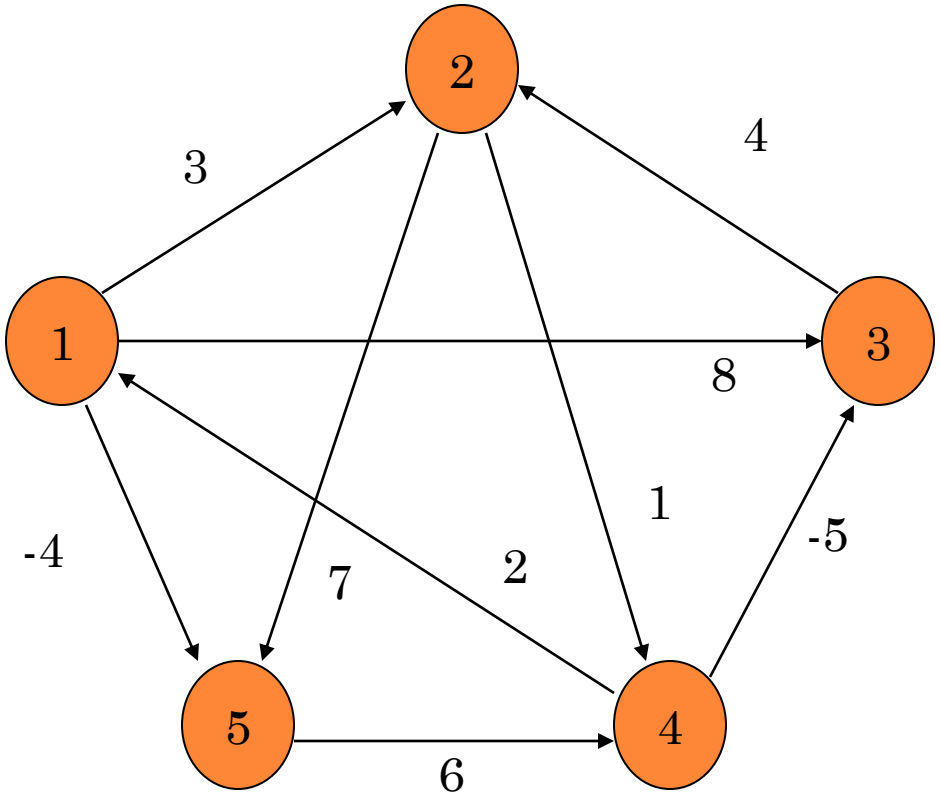
- The matrix $D^{(n)}=(d_{ij}^{(n)})$ gives the final answer- $d_{ij}^{(n)} = \delta(i, j)$ for all $i,j \in V$-because all intermediate vertices are in the set {1,2,…,n}.

ALGORITHM

```
Algorithm AllPaths(cost,A,n)
// cost[1:n,1:n] is the cost adjacency matrix of a
// graph with n vertices. A[i,j] is the cost of a
// shortest path from vertex i to vertex j. cost[i,i]=0,
// for 1<=i<=n.
{
   for i:=1 to n do
         for j:=1 to n do
               A[i,j]:=cost[i,j];
   for k:=1 to n do
         for i:=1 to n do
               for j:=1 to n do
                     A[i,j]:=min(A[i.j],A[i,k]+A[k,j];
}
```

Complexity: $O(n^3)$

Example



Solution

$$D(0)=\begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & \infty & -5 & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

$$D(1)=\begin{pmatrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & \infty & \infty \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix}$$

Solution

$$D(2)=\begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & 5 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad D(4)=\begin{pmatrix} 0 & 3 & -1 & 4 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$D(3)=\begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad D(5)=\begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

APPLICATIONS:

- Fast computation of path finder networks.
- Optimal routing.
- Detecting negative-weight cycles in graph
- Floyd- warshall algorithm is applied in electrical power system to get the shortest electrical path. Based on the information of generator group, the searching space can be narrowed by classifying load nodes.

VIVA QUESTIONS:

1. What is the purpose of Floyd's algorithm?
2. What is the time complexity of Floyd's algorithm?
3. Distinguish Bellman Ford and Floyd warshall's algortihm
4. Define optimal substructure in dynamic programming
5. Explain how John's Algorithm on All pairs shortest path is different from Floyd's Warshall's Algorithm