Dynamic Programming- All pair shortest path

Kumkum Saxena

All-Pairs Shortest Path Problem

Suppose we are given a directed graph G=(V,E) and a weight function w: E->R.

We assume that G does not contain cycles of weight 0 or less.

The All-Pairs Shortest Path Problem asks to find the length of the shortest path between any pair of vertices in G.

Quick Solutions

If the weight function is nonnegative for all edges, then we can use Dijkstra's single source shortest path algorithm for all vertices to solve the problem.

This yields an $O(n^3)$ algorithm on graphs with n vertices (on dense graphs and with a simple implementation).

Quick Solution

For arbitrary weight functions, we can use the Bellman-Ford algorithm applied to all vertices. This yields an $O(n^4)$ algorithm for graphs with n vertices.

Floyd-Warshall

We will now investigate a dynamic programming solution that solved the problem in $O(n^3)$ time for a graph with n vertices.

This algorithm is known as the Floyd-Warshall algorithm, but it was apparently described earlier by Roy.

Representation of the Input

We assume that the input is represented by a weight matrix $W = (w_{ij})_{i,j \text{ in } E}$ that is defined by

$w_{ij} = 0$ if $i=j$

$w_{ij} = w(i,j)$ if $i \neq j$ and (i,j) in E

$w_{ij} = \infty$ if $i \neq j$ and (i,j) not in E

Format of the Output

If the graph has n vertices, we return a distance matrix (d_{ij}), where d_{ij} the length of the path from i to j.

Intermediate Vertices

Without loss of generality, we will assume that V={1,2,…,n}, i.e., that the vertices of the graph are numbered from 1 to n.

Given a path p=(v_1, v_2,…, v_m) in the graph, we will call the vertices v_k with index k in {2,…,m-1} the intermediate vertices of p.

Key Definition

The key to the Floyd-Warshall algorithm is the following definition:

Let $d_{ij}^{(k)}$ denote the length of the shortest path from i to j such that all intermediate vertices are contained in the set {1,…,k}.

Remark 1

A shortest path does not contain any vertex twice, as this would imply that the path contains a cycle. By assumption, cycles in the graph have a positive weight, so removing the cycle would result in a shorter path, which is impossible.

Remark 2

Consider a shortest path p from i to j such that the intermediate vertices are from the set {1,…,k}.

- If the vertex k is not an intermediate vertex on p, then $d_{ij}^{(k)} = d_{ij}^{(k-1)}$

If the vertex k is an intermediate vertex on p, then $d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$

Interestingly, in either case, the subpaths contain merely nodes from {1,…,k-1}.

Remark 2

Therefore, we can conclude that

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)} , d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$$

Recursive Formulation

If we do not use intermediate nodes, i.e., when k=0, then

$$d_{ij}^{(0)} = w_{ij}$$

If k>0, then

$$d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)} , d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$$

The Floyd-Warshall Algorithm

```
Floyd-Warshall(W)
n = # of rows of W;
D(0) = 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)};
        od;
    od;
od;
 return D(n);
```

Time and Space Requirements

The running time is obviously $O(n^3)$.

However, in this version, the space requirements are high. One can reduce the space from $O(n^3)$ to $O(n^2)$ by using a single array d.

Dynamic programming algorithms for all-pairs shortest path

- We will study a new technique—dynamic programming algorithms (typically for optimization problems)
- **Ideas:**
  - Characterize the structure of an optimal solution
  - Recursively define the value of an optimal solution
  - Compute the value of an optimal solution in a bottom-up fashion (using matrix to compute)
  - Backtracking to construct an optimal solution from computed information.

Floyd-Warshall algorithm for 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 structure of a shortest path:

- We use a different characterization of the structure of a shortest path than we used in the matrix-multiplication-based all-pairs algorithms.
- The algorithm considers the “intermediate” vertices of a shortest path, where an intermediate vertex of a simple path $p=\langle v_1,v_2,\ldots,v_l\rangle$ is any vertex in p other than v_1 or v_l, that is, any vertex in the set $\{v_2,v_3,\ldots,v_{l-1}\}$

Continue:

- 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 $\in$ 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}.

Relationship:

- 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.p1 is a shortest path from i to k with all intermediate vertices in the set {1,2,…,k-1}, so as p2.

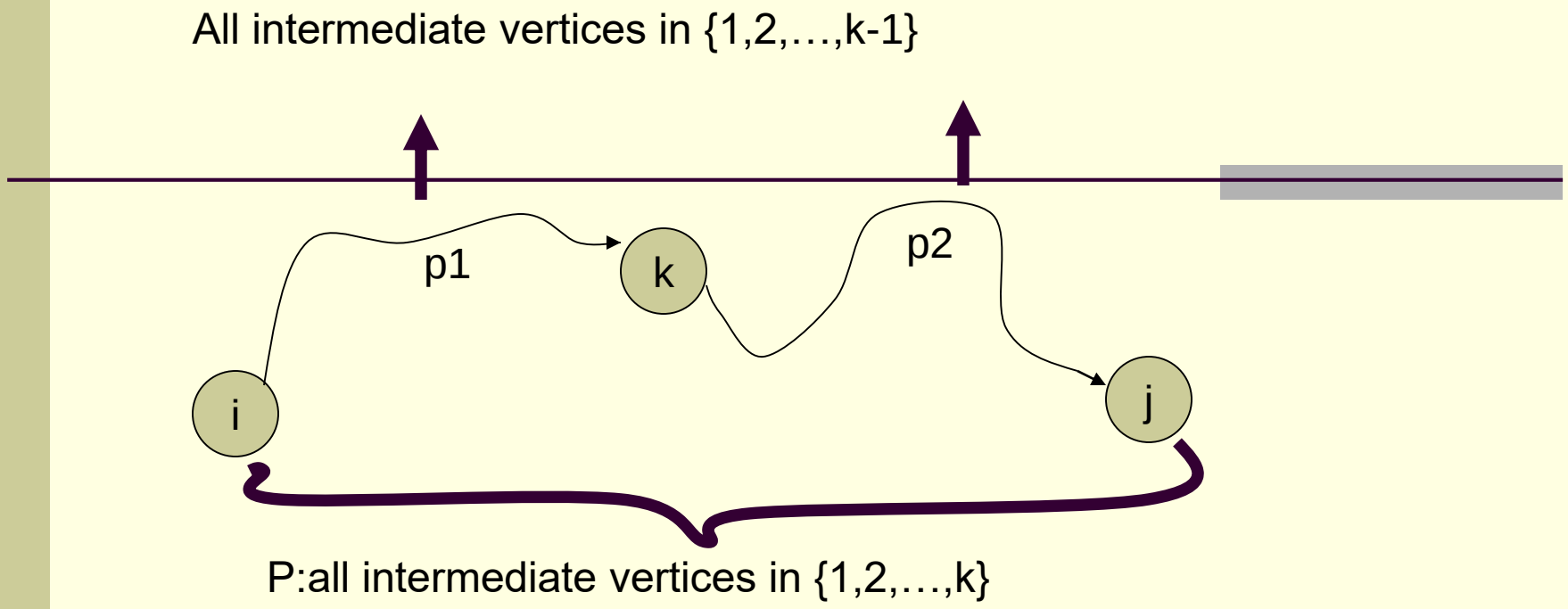


Figure 2. Path p is a shortest path from vertex i to vertex j,and k is the highest-numbered intermediate vertex of p. Path p1, the portion of path p from vertex i to vertex k,has all intermediate vertices in the set {1,2,…,k-1}.The same holds for path p2 from vertex k to vertex j.

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
- $d_{ij}^{(k)} = \begin{cases} w_{ij} & \geq \text{ if } k=0, \\ \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}) & \text{if } k \quad 1. \end{cases}$

- 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}.

Computing the shortest-path weights bottom up:

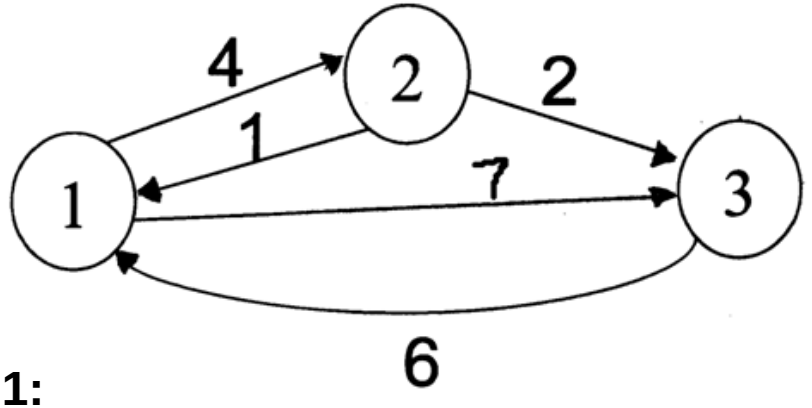
- FLOYD-WARSHALL(W)
- n $\leftarrow$ rows[W]
- $D^{(0)} \leftarrow W$
- for k $\leftarrow$ 1 to n
- do for i $\leftarrow$ 1 to n
- do for j $\leftarrow$ 1 to n
- $d_{ij}^{(k)} \leftarrow \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$

- return $D^{(n)}$

Floyd Warshall Algorithm - Example

$$D^{(0)} = \begin{bmatrix} 0 & 4 & 7 \\ 1 & 0 & 2 \\ 6 & \infty & 0 \end{bmatrix}$$

Original weights.



$$D^{(1)} = \begin{bmatrix} 0 & 4 & 7 \\ 1 & 0 & 2 \\ 6 & 10 & 0 \end{bmatrix}$$

Consider Vertex 1:

D(3,2) = D(3,1) + D(1,2)

$$D^{(2)} = \begin{bmatrix} 0 & 4 & 6 \\ 1 & 0 & 2 \\ 6 & 10 & 0 \end{bmatrix}$$

Consider Vertex 2:

D(1,3) = D(1,2) + D(2,3)

$$D^{(3)} = \begin{bmatrix} 0 & 4 & 6 \\ 1 & 0 & 2 \\ 6 & 10 & 0 \end{bmatrix}$$

Consider Vertex 3:

Nothing changes.

Floyd Warshall Algorithm

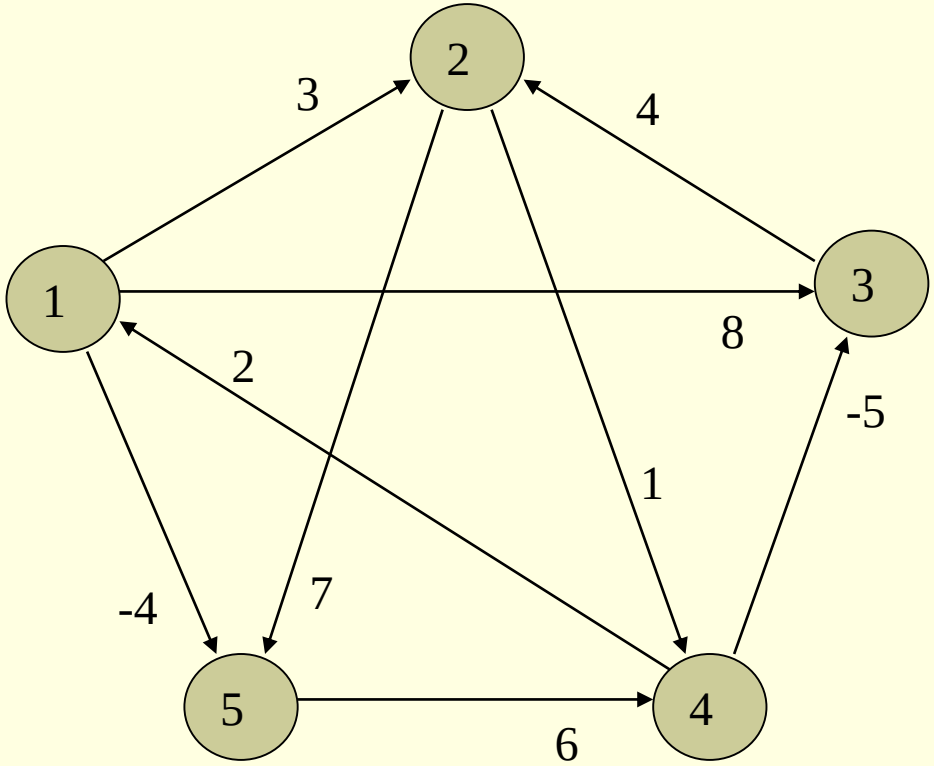
- Looking at this example, we can come up with the following algorithm:
  - Let D store the matrix with the initial graph edge information initially, and update D with the calculated shortest paths.

```
For k=1 to n {
  For i=1 to n {
          For j=1 to n
             D[i,j] = min(D[i,j],D[i,k]+D[k,j])
  }
}
```

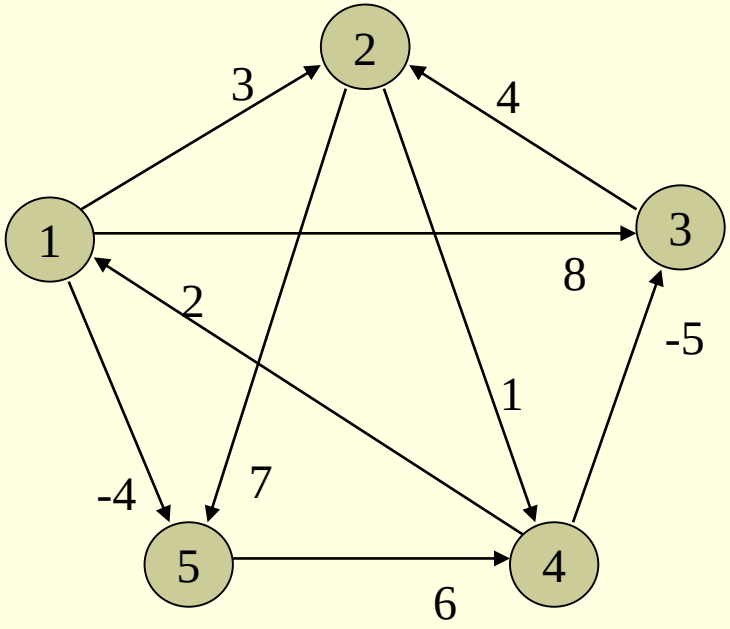
  - The final D matrix will store all the shortest paths.

Shortest Paths and Matrix Multiplication

- Example



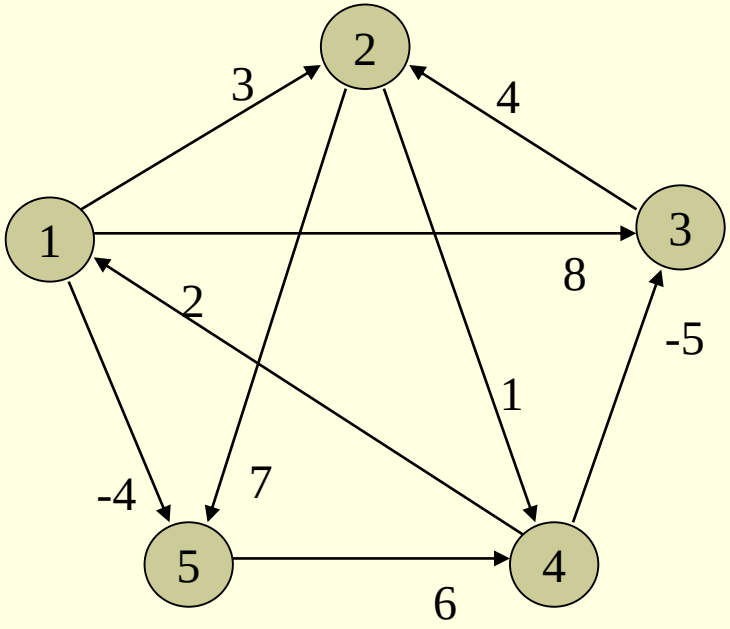
Shortest Paths and Matrix Multiplication



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

$D^1 = D^0 W$

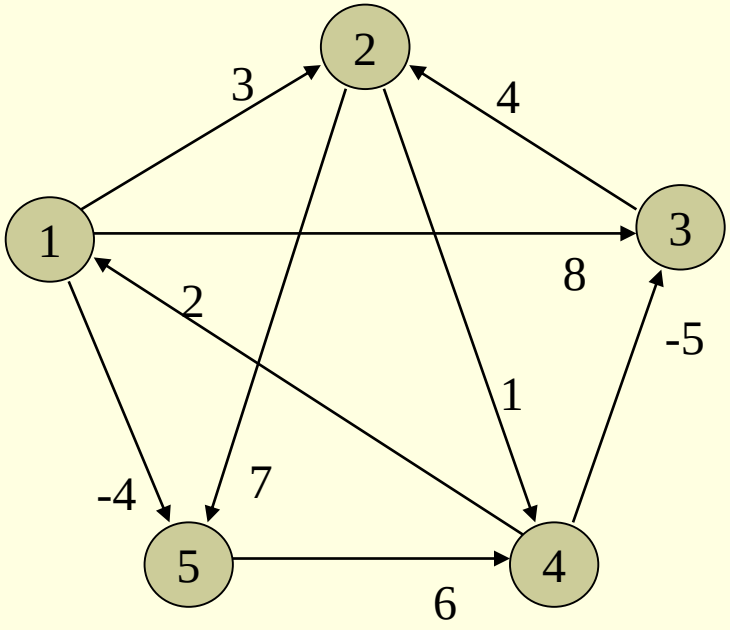
Shortest Paths and Matrix Multiplication



	1	2	3	4	5
1	0	3	8	2	-4
2	3	0	-4	1	7
3	∞	4	0	5	11
4	2	-1	-5	0	-2
5	8	∞	1	6	0

$D^2 = D^1 W$

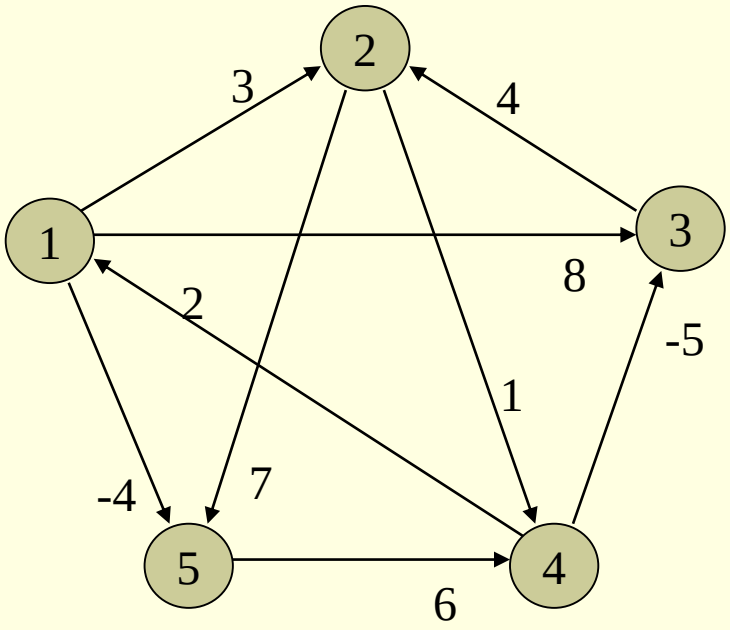
Shortest Paths and Matrix Multiplication



	1	2	3	4	5
1	0	3	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	11
4	2	-1	-5	0	-2
5	8	5	1	6	0

$D^3 = D^2 W$

Shortest Paths and Matrix Multiplication

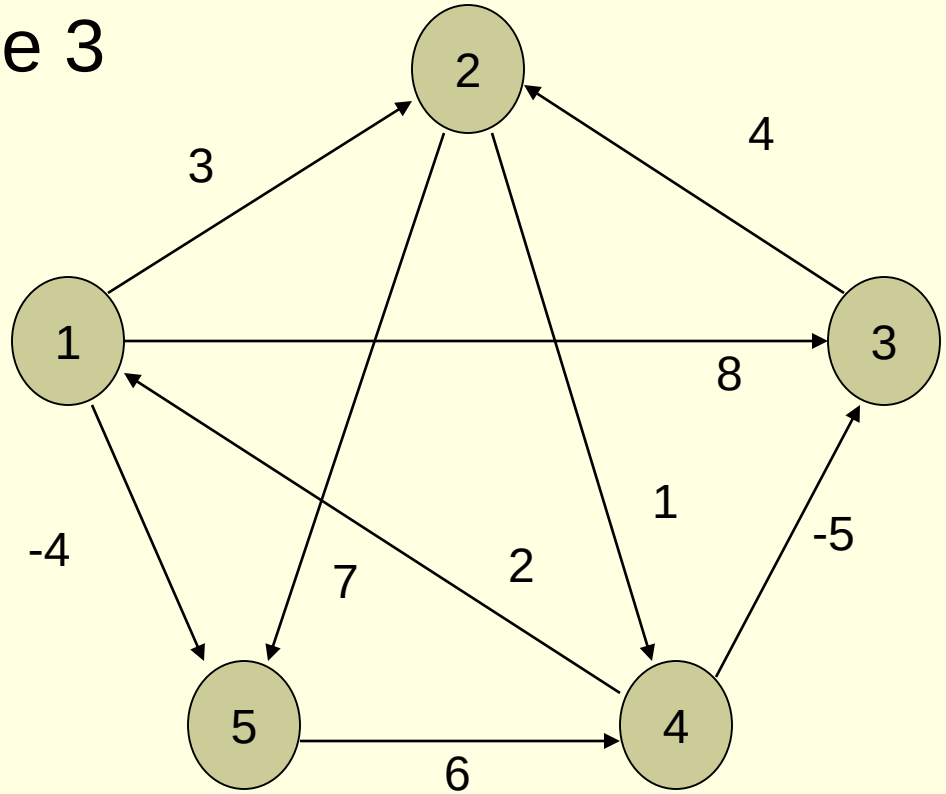


	1	2	3	4	5
1	0	1	-3	2	-4
2	3	0	-4	1	-1
3	7	4	0	5	3
4	2	-1	-5	0	-2
5	8	5	1	6	0

$D^4 = D^3 W$

Example:

- Figure 3



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

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

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

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

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

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

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} \text{ if } d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} \text{ if } d_{ij}^{(k-1)} > d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \end{cases}$$

Step (i) When k = 0

$$D^{(0)} = \begin{matrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & -5 & \infty \\ 2 & \infty & \infty & 0 & \infty \\ \infty & \infty & \infty & 6 & 0 \end{matrix} \qquad \pi^{(0)} = \begin{matrix} \text{NIL} & 1 & 1 & \text{NIL} & 1 \\ \text{NIL} & \text{NIL} & \text{NIL} & 2 & 2 \\ \text{NIL} & 3 & \text{NIL} & 3 & \text{NIL} \\ 4 & \text{NIL} & \text{NIL} & \text{NIL} & \text{NIL} \\ \text{NIL} & \text{NIL} & \text{NIL} & 5 & \text{NIL} \end{matrix}$$

Step (ii) When k =1

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

$$d_{14}^{(1)} = \min\left(d_{14}^{(0)}, d_{11}^{(0)} + d_{14}^{(0)}\right)$$

$$d_{14}^{(1)} = \min(\infty, 0 + \infty) = \infty$$

$$d_{15}^{(1)} = \min\left(d_{15}^{(0)}, d_{11}^{(0)} + d_{15}^{(0)}\right)$$

$$d_{15}^{(1)} = \min(-4, 0 + -4) = -4$$

$$d_{21}^{(1)} = \min\left(d_{21}^{(0)}, d_{21}^{(0)} + d_{11}^{(0)}\right)$$

$$d_{21}^{(1)} = \min(\infty, \infty + 0) = \infty$$

$$d_{23}^{(1)} = \min\left(d_{23}^{(0)}, d_{21}^{(0)} + d_{13}^{(0)}\right)$$

$$d_{23}^{(1)} = \min((\infty, \infty + 8) = \infty$$

$$d_{31}^{(1)} = \min\left(d_{31}^{(0)}, d_{31}^{(0)} + d_{11}^{(0)}\right)$$

$$d_{31}^{(1)} = \min(\infty, \infty + 0) = \infty$$

$$d_{35}^{(1)} = \min (d_{35}^{(0)}, d_{31}^{(0)} + d_{15}^{(0)})$$

$$d_{35}^{(1)} = \min (\infty, \infty + (-4)) = \infty$$

$$d_{42}^{(1)} = \min (d_{42}^{(0)}, d_{41}^{(0)} + d_{12}^{(0)})$$

$$d_{42}^{(1)} = \min (\infty, 2 + 3) = 5$$

$$d_{43}^{(1)} = \min (d_{43}^{(0)}, d_{41}^{(0)} + d_{13}^{(0)})$$

$$d_{43}^{(1)} = \min (\infty, 2 + 8) = 10$$

$$d_{45}^{(1)} = \min (d_{45}^{(0)}, d_{41}^{(0)} + d_{15}^{(0)})$$

$$d_{45}^{(1)} = \min (\infty, 2 + (-4)) = -2$$

$$d_{51}^{(1)} = \min (d_{51}^{(0)}, d_{51}^{(0)} + d_{11}^{(0)})$$

$$d_{51}^{(1)} = \min (\infty, \infty + 0) = \infty$$

$$D_{ij}^{(1)} = \begin{matrix} 0 & 3 & 8 & \infty & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & -5 & \infty \\ 2 & 5 & 10 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{matrix} \qquad \pi^{(1)} = \begin{matrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 3 & NIL \\ 4 & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{matrix}$$

Step (iii) When k = 2

$$d_{ij}^{(k)} = \min (d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$$

$$d_{14}^{(2)} = \min (d_{14}^{(1)}, d_{12}^{(1)} + d_{24}^{(1)})$$

$$d_{14}^{(2)} = \min (\infty, 3 + 1) = 4$$

$$d_{21}^{(2)} = \min (d_{21}^{(1)}, d_{22}^{(1)} + d_{21}^{(1)})$$

$$d_{21}^{(2)} = \min (\infty , 0 + \infty) = \infty$$

$$d_{34}^{(2)} = \min (d_{34}^{(1)}, d_{32}^{(1)} + d_{24}^{(1)})$$

$$d_{34}^{(2)} = \min (-5, 4 + 1) = -5$$

$$d_{35}^{(2)} = \min (d_{35}^{(1)}, d_{32}^{(1)} + d_{25}^{(1)})$$

$$d_{35}^{(2)} = \min (\infty, 4 + 7) = 11$$

$$d_{43}^{(2)} = \min (d_{43}^{(1)}, d_{42}^{(1)} + d_{23}^{(1)})$$

$$d_{43}^{(2)} = \min (10, 5 + \infty) = 10$$

$$D^{(2)}_{ij} = \begin{bmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & -5 & 11 \\ 2 & 5 & 10 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{bmatrix} \qquad \pi^{(2)} = \begin{bmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 3 & 2 \\ 4 & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{bmatrix}$$

Step (iv) When k = 3

$$d_{ij}^{(k)} = \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$$

$$d_{14}^{(3)} = \min\left(d_{14}^{(2)}, d_{13}^{(2)} + d_{34}^{(2)}\right)$$

$$d_{14}^{(3)} = \min(4, 8 + (-5)) = 3$$

$$D_{ij}^{(3)} = \begin{matrix} 0 & 3 & 8 & 3 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & -5 & 11 \\ 2 & 5 & 10 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{matrix} \qquad \pi^{(3)} = \begin{matrix} NIL & 1 & 1 & 3 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 3 & 2 \\ 4 & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{matrix}$$

$\infty \quad \infty \quad \infty \quad 6 \quad 0$

Step (v) When k = 4

$$d_{ij}^{(k)} = \min (d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$$

$$d_{21}^{(4)} = \min (d_{21}^{(3)}, d_{24}^{(3)} + d_{41}^{(3)})$$

$$d_{21}^{(4)} = \min (\infty, 1 + 2) = 3$$

$$d_{23}^{(4)} = \min (d_{23}^{(3)}, d_{24}^{(3)} + d_{43}^{(3)})$$

$$d_{23}^{(4)} = \min (\infty, 1 + 10) = 11$$

$$d_{25}^{(4)} = \min (d_{25}^{(3)}, d_{24}^{(3)} + d_{45}^{(3)})$$

$$d_{25}^{(4)} = \min (7, 1 + (-2)) = -1$$

$$d_{31}^{(4)} = \min (d_{31}^{(3)}, d_{34}^{(3)} + d_{41}^{(3)})$$

$$d_{31}^{(4)} = \min (\infty, -5 + 2) = -3$$

$$d_{32}^{(4)} = \min (d_{32}^{(3)}, d_{34}^{(3)} + d_{42}^{(3)})$$

$$d_{32}^{(4)} = \min (4, -5 + 5) = 0$$

$$d_{32}^{(4)} = \min (d_{32}^{(3)}, d_{34}^{(3)} + d_{42}^{(3)})$$

$$d_{32}^{(4)} = \min (4, -5 + 5) = 0$$

$$d_{51}^{(4)} = \min (d_{51}^{(3)}, d_{54}^{(3)} + d_{41}^{(3)})$$

$$d_{51}^{(4)} = \min (\infty, 6 + 2) = 8$$

$$d_{52}^{(4)} = \min (d_{52}^{(3)}, d_{54}^{(3)} + d_{42}^{(3)})$$

$$d_{52}^{(4)} = \min (\infty, 6 + 5) = 11$$

$$d_{53}^{(4)} = \min (d_{53}^{(3)}, d_{54}^{(3)} + d_{43}^{(3)})$$

$$d_{53}^{(4)} = \min (\infty, 6 + 10) = 16$$

$$D_{ij}^{(4)} = \begin{matrix} 0 & 3 & 8 & 3 & -4 \\ 3 & 0 & 11 & 1 & -1 \\ -3 & 0 & 0 & -5 & -7 \\ 2 & 5 & 10 & 0 & -2 \\ 8 & 11 & 16 & 6 & 0 \end{matrix} \qquad \pi^{(4)} = \begin{matrix} NIL & 1 & 1 & 3 & 1 \\ 4 & NIL & 4 & 2 & 2 \\ 4 & 4 & NIL & 3 & 4 \\ 4 & 1 & 1 & NIL & 1 \\ 4 & 4 & 4 & 5 & NIL \end{matrix}$$

Step (vi) When k = 5

$$d_{ij}^{(k)} = \min (d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$$

$$d_{25}^{(5)} = \min (d_{25}^{(4)}, d_{25}^{(4)} + d_{55}^{(3)})$$

$$d_{25}^{(5)} = \min (-1, -1 + 0) = -1$$

$$d_{23}^{(5)} = \min (d_{23}^{(4)}, d_{25}^{(4)} + d_{53}^{(3)})$$

$$d_{23}^{(5)} = \min (11, -1 + 16) = 11$$

$$d_{35}^{(5)} = \min (d_{35}^{(4)}, d_{35}^{(4)} + d_{55}^{(3)})$$

$$d_{35}^{(5)} = \min (-7, -7 + 0) = -7$$

$$D_{ij}^{(5)} = \begin{matrix} 0 & 3 & 8 & 3 & -4 \\ 3 & 0 & 11 & 1 & -1 \\ -3 & 0 & 0 & -5 & -7 \\ 2 & 5 & 10 & 0 & -2 \\ 8 & 11 & 16 & 6 & 0 \end{matrix} \qquad \pi^{(5)} = \begin{matrix} NIL & 1 & 1 & 5 & 1 \\ 4 & NIL & 4 & 2 & 4 \\ 4 & 4 & NIL & 3 & 4 \\ 4 & 1 & 1 & NIL & 1 \\ 4 & 4 & 4 & 5 & NIL \end{matrix}$$

Applications

- To automatically find directions between physical locations.
- Vehicle Routing and scheduling
- In a networking or telecommunication applications, Dijkstra's algorithm has
- been used for solving the min-delay path problem (which is the shortest path problem).
- For example in data network routing, the goal is to find the path for data
- packets to go through a switching network with minimal delay.