All Pairs Shortest Paths
Book: Cormen; Chapter: 25

Floyd-Warshall : Section: 25.2

All-Pairs Shortest Paths Problem

- Problem:

  Given a directed graph $G=(V, E)$, and a weight function $w: E \rightarrow \mathbb{R}$, for each pair of vertices u, v, compute the shortest path weight $\delta(u, v)$, and a shortest path if exists.

- Output:
  - A $V \times V$ matrix $D = (d_{ij})$, where, d_{ij} contains the shortest path weight from vertex i to vertex j.
  - A $V \times V$ matrix $\Pi=(\pi_{ij})$, where, π_{ij} is NIL if either $i=j$ or there is no path from i to j, otherwise π_{ij} is the predecessor of j on some shortest path from i.

Adjacency Matrix: $W = (w_{ij})$

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

Methods

- Single-source shortest-path algorithms
- Direct methods
  - Matrix multiplication
  - **Floyd-Warshall Algorithm**
  - Johnson's Algorithm for sparse graphs
- Transitive closure (Floyd-Warshall Algorithm)

Floyd-Warshall Algorithm
Structure of a Shortest Path

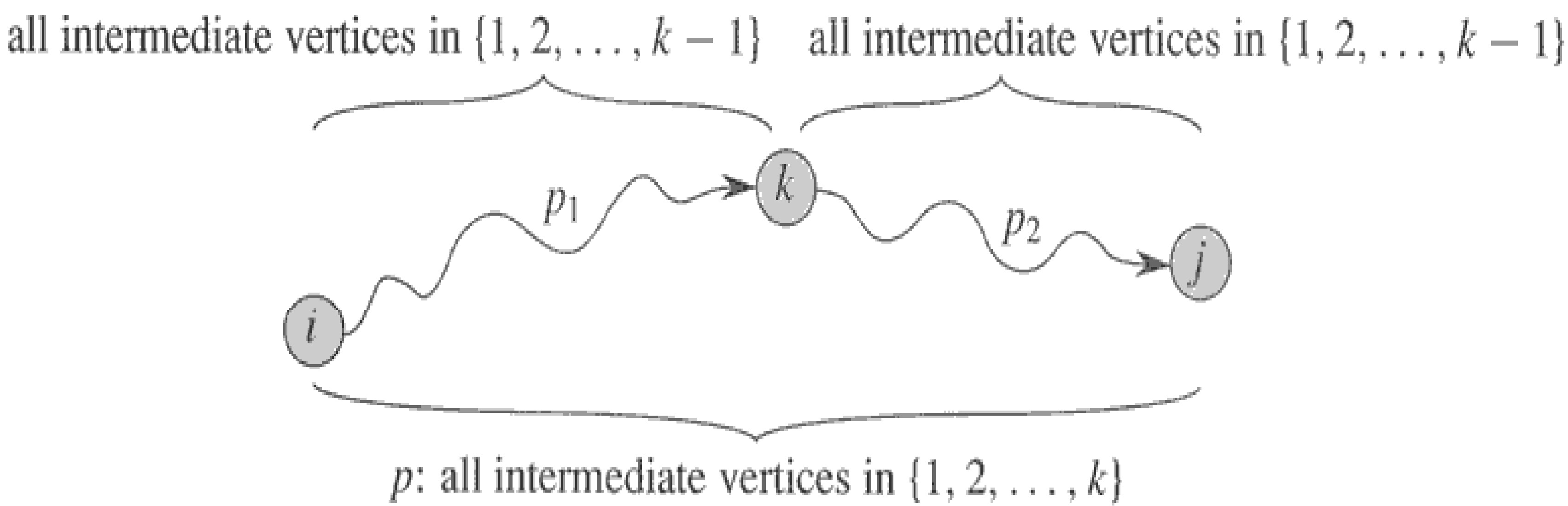
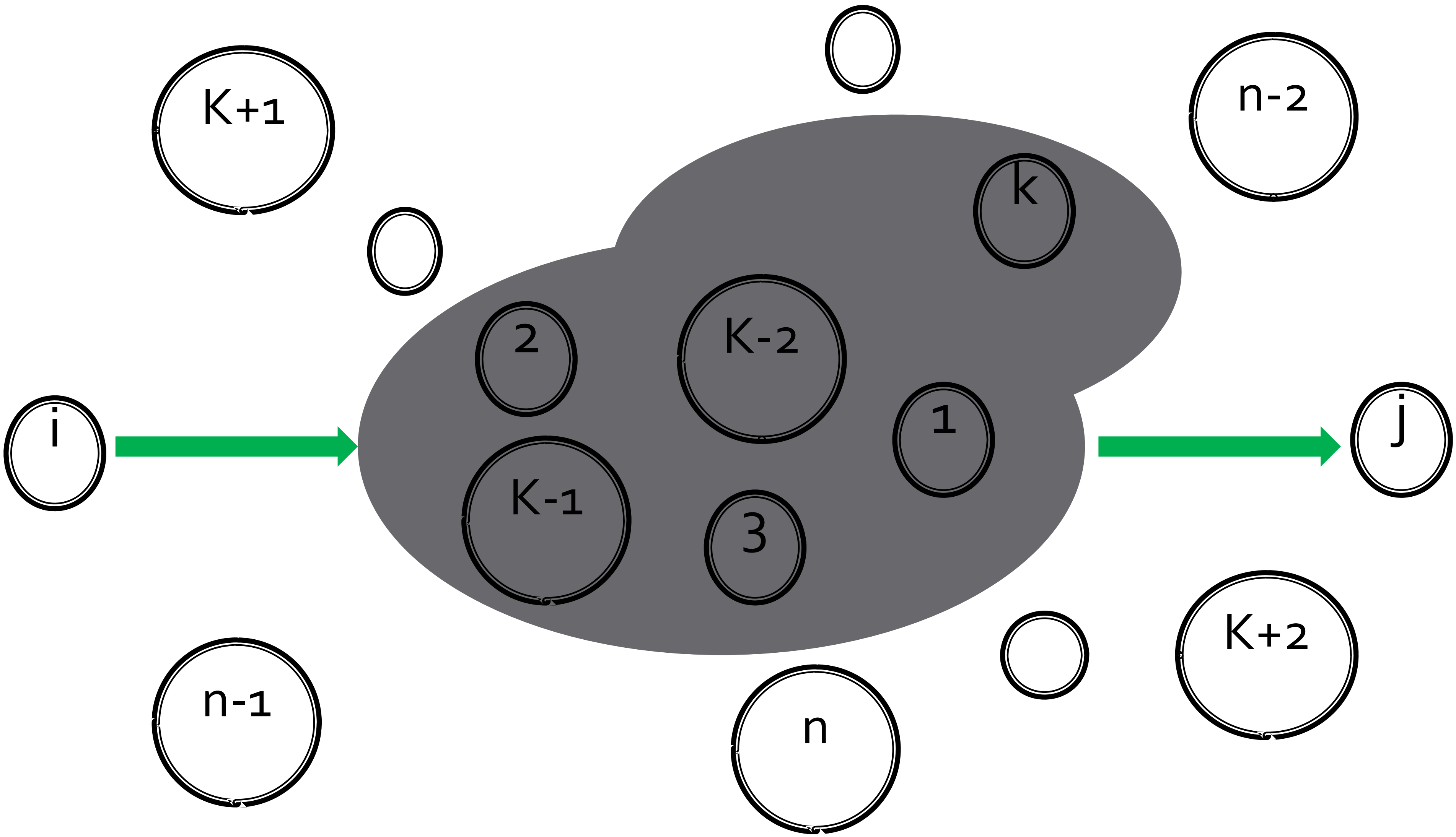


Figure 25.3 Path p is a shortest path from vertex i to vertex j, and k is the highest-numbered intermediate vertex of p. Path p_1, the portion of path p from vertex i to vertex k, has all intermediate vertices in the set $\{1, 2, \ldots, k-1\}$. The same holds for path p_2 from vertex k to vertex j.

Floyd-Warshall Algorithm
Structure of a Shortest Path (Explanation)



Floyd-Warshall Algorithm
Recursive Solution

- $d_{ij}^{(k)}$:
  shortest path weight from *i* to *j*
  with intermediate vertices **(excluding *i, j*)**
  from the set $\{1,2,\ldots,k\}$

- $d_{ij}^{(0)} = w_{ij}$;
- $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$; if $k \geq 1$

Result: $D^{(n)} = (d_{ij}^{(n)})$

Constructing a Shortest Path

- For $k = 0$

$$\pi_{ij}^{(0)} = \begin{cases} NIL & \text{if } i = j \text{ or } w_{ij} = \infty \\ i & \text{if } i \neq j \text{ and } w_{ij} < \infty \end{cases}$$

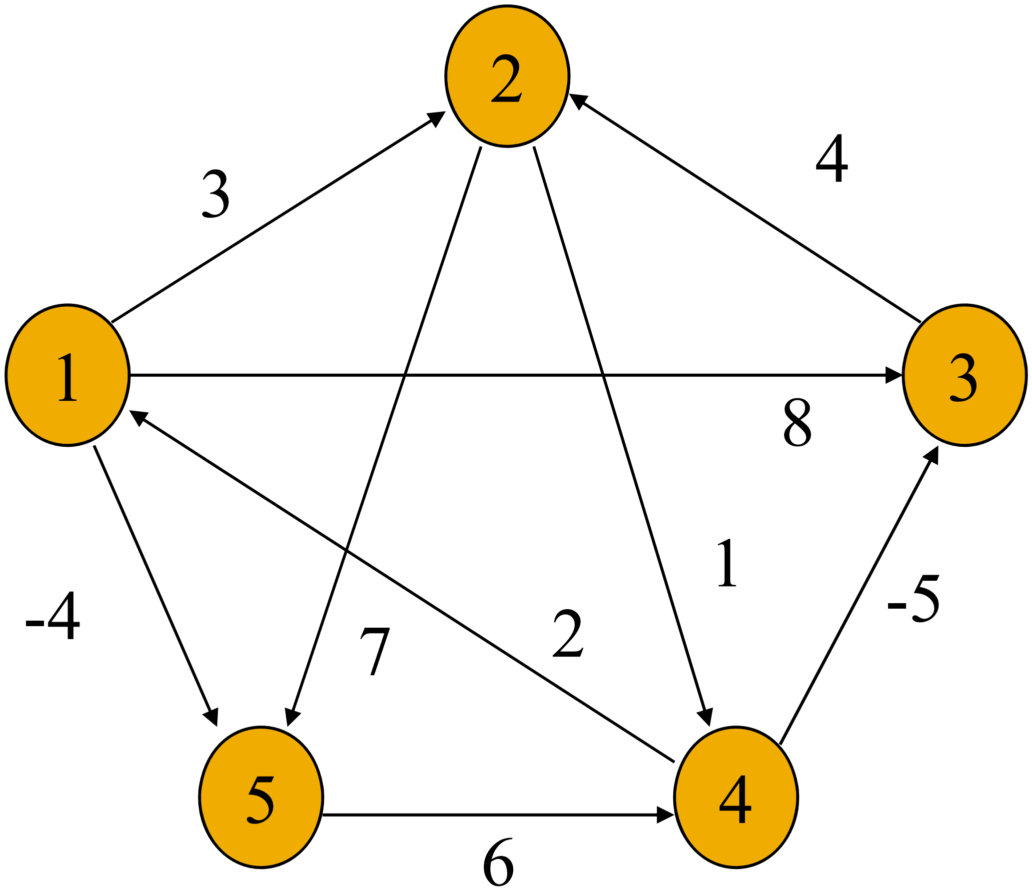
- For $k \geq 1$

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

Compute Weights and Construct Paths

- Solve the problem stage by stage:
  - $D(0)$, $\Pi(0)$
  - $D^{(1)}$, $\Pi^{(1)}$
  - $D^{(2)}$, $\Pi^{(2)}$
  - …, …
  - $D^{(n)}$, $\Pi^{(n)}$
- where $D^{(k)}$ contains the shortest path weight with all the **intermediate vertices** from set $\{1,2\ldots,k\}$.

Example (Page - 690)



Example (Page - 696)

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

$$\prod^{(0)}=\begin{pmatrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & NIL & NIL \\ 4 & NIL & 4 & NIL & NIL \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

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

$$\prod^{(1)}=\begin{pmatrix} NIL & 1 & 1 & NIL & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & NIL & NIL \\ 4 & 1 & 4 & NIL & 1 \\ NIL & NIL & NIL & 5 & NIL \end{pmatrix}$$

Example (Page - 696)

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

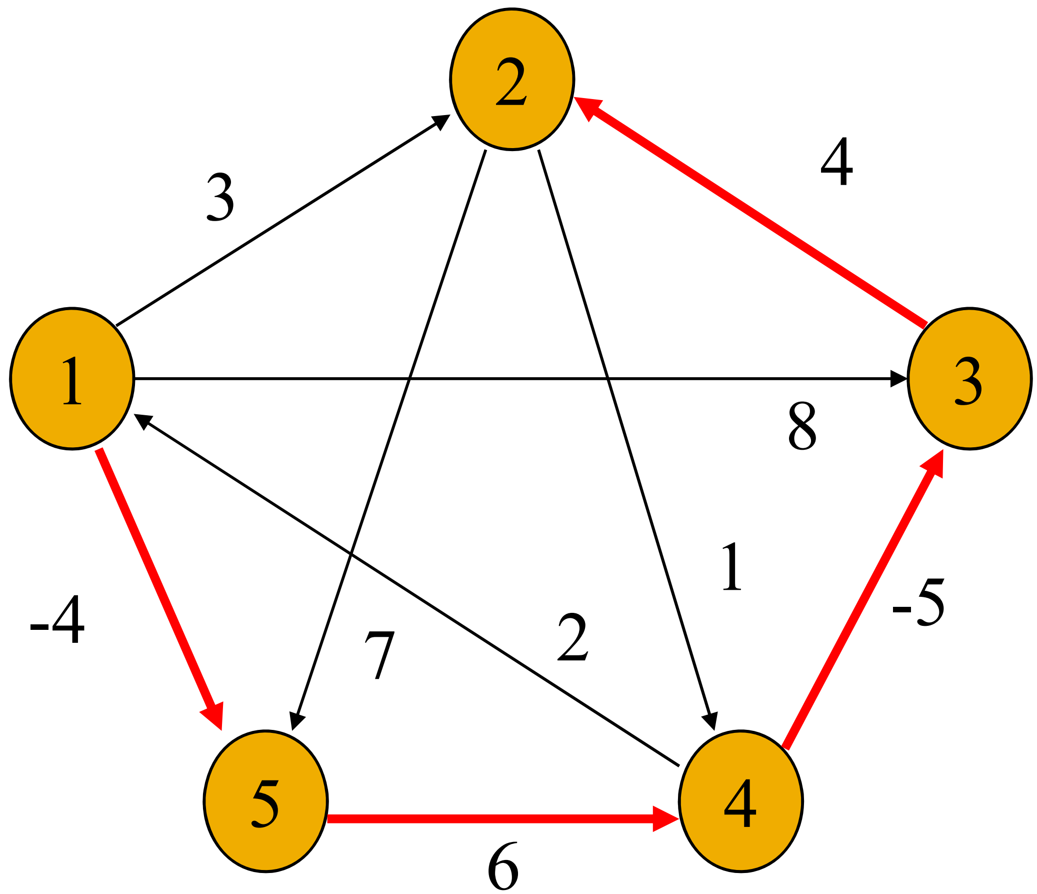
$$D^{(3)}= \begin{pmatrix} 0 & 3 & 8 & 4 & -4 \\ \infty & 0 & \infty & 1 & 7 \\ \infty & 4 & 0 & 5 & 11 \\ 2 & -1 & -5 & 0 & -2 \\ \infty & \infty & \infty & 6 & 0 \end{pmatrix} \qquad \prod^{(3)}= \begin{pmatrix} NIL & 1 & 1 & 2 & 1 \\ NIL & NIL & NIL & 2 & 2 \\ NIL & 3 & NIL & 2 & 2 \\ 4 & 3 & 4 &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

Example (Page - 696)

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

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

Shortest Path from 1 to 2



Algorithm (Page - 695)

FLOYD-WARSHALL(W)

$n = W.rows$

$D^{(0)} = W$

for $k = 1$ **to** n

  let $D^{(k)} = (d_{ij}^{(k)})$ be a new $n \times n$ matrix

  for $i = 1$ **to** n

    for $j = 1$ **to** n

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

return $D^{(n)}$

Print Shortest Paths (Page - 685)

```
PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, j)
if i = j
    then print i
    else if π_ij = NIL
            then print "no path from" i "to" j "exists"
            else PRINT-ALL-PAIRS-SHORTEST-PATH(Π, i, π_ij)
                 print j
```

Thank You

Stay Safe