All-Pairs Shortest Path

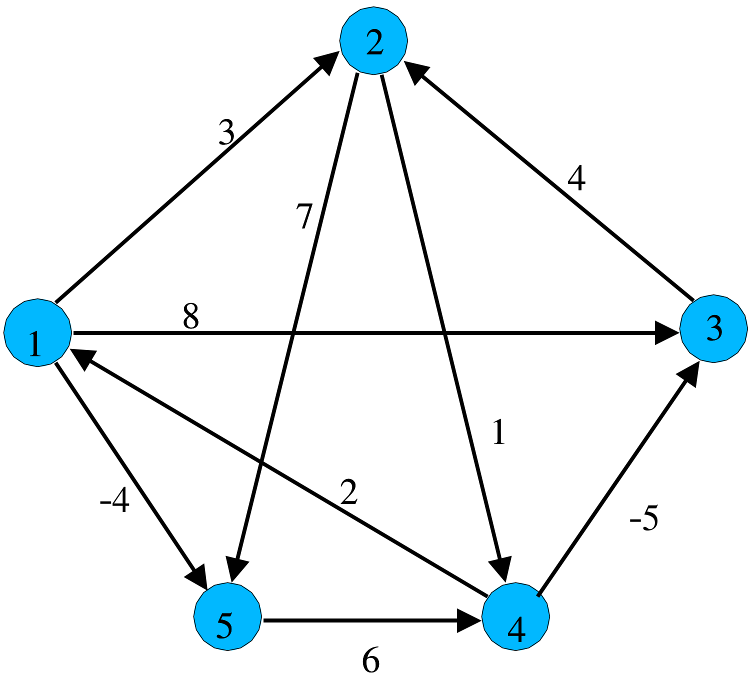
Dr. Gur Saran Adhar

Reference clrs, Chapter 25, Page 620-

Introduction to Problem:
All Pairs Shortest Paths

`Reference: clrs, page-620`

Given a **weighted, directed graph** $G = (V, E)$, with weight function $w : E \to R$ mapping edges to real-values weights. Find shortest paths between **all pairs** of vertices.
This problem arises in making a table of distances between all pairs of cities for a road atlas. Typically, the table in row u and column v is the length of the shortest path from u to v.



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

Representation

We assume that the vertices are numbered $1, 2, \ldots, |V|$.

The **input** is an $n \times n$ matrix W representing the edge weights on an n vertex directed graph $G = (V, E)$ where:

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

The **output** is the **predecessor matrix** $\Pi = (\pi_{ij})$

$$\pi_{ij} = \begin{cases} \text{NIL} & \text{if } i = j \text{ or there is no shortest pat} \\ \text{predecessor of } j & \end{cases}$$

Main Idea

Note: Induction on the number of edges in the shortest path.

Let l_{ij}^m denotes the length of shortest path from i to j **that contains at most** m **edges**.

$$l_{ij}^0 = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{if } i \neq j \end{cases}$$

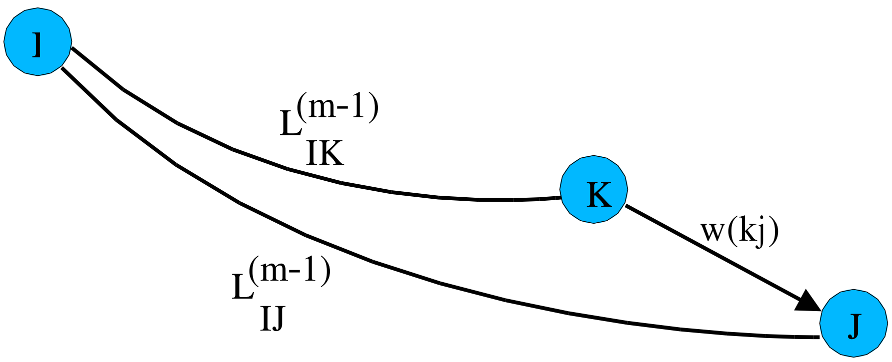
For $m \geq 1$, we compute l_{ij}^m by taking the minimum of l_{ij}^{m-1} (the weight of the shortest path from i to j containing at most $m-1$ edges) and the minimum weight of path with m edges (obtained by looking at all predecessors of j)

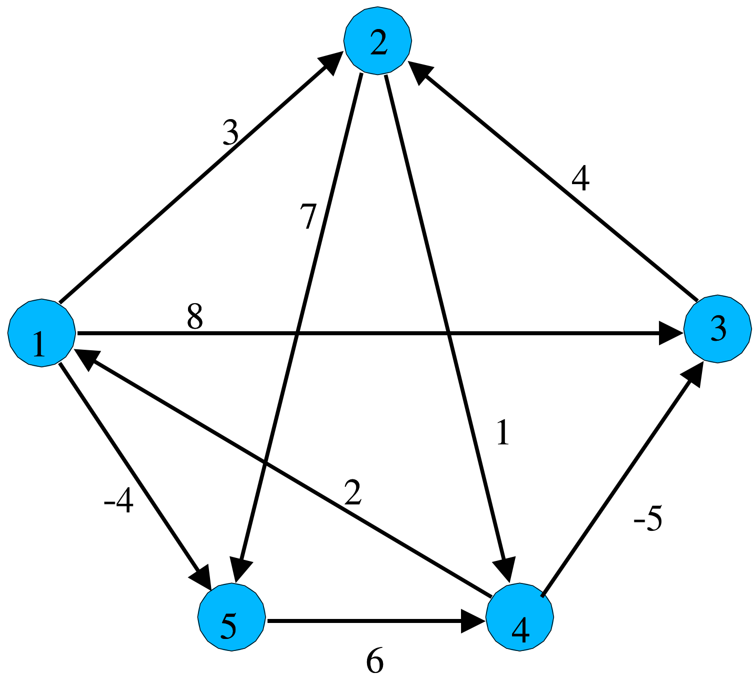
$$\begin{aligned} l_{ij}^m &= \min(l_{ij}^{m-1}, \min_{1 \leq k \leq n}\{l_{ik}^{m-1} + w_{kj}\}) \\ &= \min_{1 \leq k \leq n}(l_{ik}^{m-1} + w_{kj}\} \end{aligned}$$

If the graph contains no negative weight cycles, then for every pair of vertices i **and** j **there is a shortest path that contains at most** $(n-1)$ **edges**.

The **weight** of shortest path are therefore,

$$\delta(i,j) = l_{ij}^{n-1} = l_{ij}^n = l_{ij}^{n+1} = \ldots$$


I
K
J
L (m-1) IK
L (m-1) IJ
w(kj)



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

Example

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

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

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

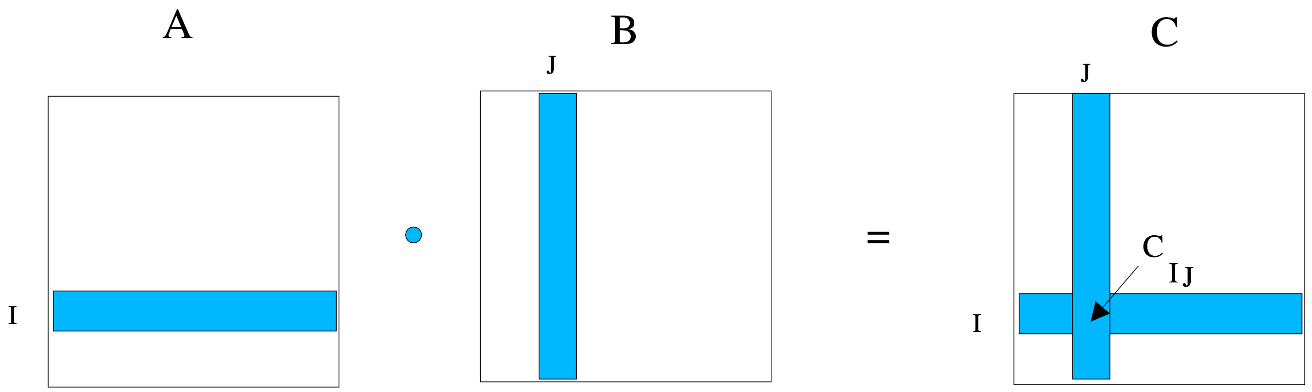
$$L^{(4)} = \begin{pmatrix} 0 & 1 & -3 & 2 & -4 \\ 3 & 0 & -4 & 1 & -1 \\ 7 & 4 & 0 & 5 & 3 \\ 2 & -1 & -5 & 0 & -2 \\ 8 & 5 & 1 & 6 & 0 \end{pmatrix}$$

$$c_{ij} = \sum_{k=1}^{n} a_{ik}.b_{kj}$$

MATRIX-MULTIPLY(A, B)

```
n ← rows[L]
let C be an n × n matrix
for i ← 1 to n
    do for j ← 1 to n
        do c_ij = 0
               for k ← 1 to n
                   do c_ij ← c_ij + a_ik.b_kj
return C
```

Reference clrs 625


A
B
C
I
J
J
=
C
IJ
I

EXTEND-SHORTEST-PATHS(L, W)

$n \leftarrow rows[L]$
let $L^{'} = (l^{'}_{ij})$ be an $n \times n$ matrix
for $i \leftarrow 1$ to n
    do for $j \leftarrow 1$ to n
        do $l^{'}_{ij} = infty$
            for $k \leftarrow 1$ to n
                do $l^{'}_{ij} = \min\{l^{'}_{ij}, l^{'}_{ik} + w_{kj}\}$
return $L^{'}$

`Reference clrs 624`

$$
\begin{array}{lll}
L^1 & = L^0.W & = W \\
L^2 & = L^1.W & = W^2 \\
L^3 & = L^2.W & = W^3 \\
\vdots & & \\
L^{n-1} & = L^{n-2}.W & = W^{n-1}
\end{array}
$$

ALL-PAIRS-SHORTEST-PATHS(W)

```
n ← rows[W]
L^1 ← W
for m ← 2 to n − 1
    do L^m ← EXTEND-SHORTEST-PATHS(L^(m−1), W)
return L^(n−1)
```

Reference clrs 625

$$
\begin{aligned}
L^1 &= L^0.W &&= W \\
L^2 &= W^2 &&= W.W \\
L^4 &= W^4 &&= W^2.W^2 \\
L^8 &= W^8 &&= W^4.W^4 \\
&\vdots
\end{aligned}
$$

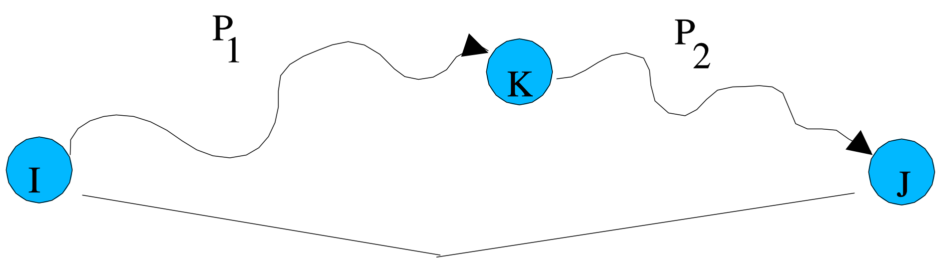
FAST-PAIRS-SHORTEST-PATHS(W)

```
n ← rows[W]
L^1 ← W
m ← 1
while m < n − 1
    do  L^{2m} ← EXTEND-SHORTEST-PATHS(L^m, L^m)
return L^m
```

Reference clrs 627

Floyd Warshall Algorithm: Main Idea

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


P1
K
P2
I
J
P: all intermediate vertices in {1,2,....k}

FLOYD-WARSHALL(W)

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

Reference clrs 624