All Pairs Shortest Paths
(Floyd Warshall)

All Pairs Shortest Paths (APSP)

- given : directed graph $G = (V, E)$,$|V| = n$
- goal : create an $n \times n$ matrix $D = (d_{ij})$ of shortest path distances
  i.e., $d_{ij} = \delta(v_i, v_j)$
- Solution : run a **Single Source Shortest Pair** algorithm n times, one for each vertex as the source.

Adjacency Matrix Representation of Graphs

►n x n matrix W = (ω_{ij}) of edge weights :

$$\omega_{ij} = \begin{cases} \omega(v_i, v_j) & \text{if } (v_i, v_j) \in E \\ \infty & \text{if } (v_i, v_j) \notin E \end{cases}$$

►assume $\omega_{ii} = 0$ for all $v_i \in V$, because

– no neg-weight cycle

$\Rightarrow$ shortest path to itself has no edge,

i.e., $\delta(v_i, v_i) = 0$

Dynamic Programming

(1) Characterize the structure of an optimal solution.

(2) Recursively define the value of an optimal solution.

(3) Compute the value of an optimal solution in a bottom-up manner.

(4) Construct an optimal solution from information constructed in (3).

Shortest Paths

Assumption : negative edge weights may be present, but no negative weight cycles.

(1) Structure of a Shortest Path :

- Consider a shortest path p_{ij}^{m} from v_i to v_j such that $|p_{ij}^{m}| \le m$
  - ▶ i.e., path p_{ij}^{m} has at **most *m* edges.**
- no negative-weight cycle $\Rightarrow$ all shortest paths are simple
  - $\Rightarrow$ m is finite $\Rightarrow m \le n - 1$
- $i = j \Rightarrow |p_{ii}| = 0$ & $\omega(p_{ii}) = 0$
- $i \ne j \Rightarrow$ decompose path p_{ij}^{m} into $\mathbf{p_{ik}^{m-1}}$ **&** $\mathbf{v_k \rightarrow v_j}$, where $|p_{ik}^{m-1}| \le m - 1$
  - ▶ p_{ik}^{m-1} should be a shortest path from v_i to v_k by optimal substructure property.
  - ▶ Therefore, $\boldsymbol{\delta(v_i, v_j) = \delta(v_i, v_k) + \omega_{kj}}$

Shortest Paths

(2) A Recursive Solution to All Pairs Shortest Paths Problem :

- d_{ij}^{m} = minimum weight of any path from v_i to v_j that contains at most "*m*" edges.
- $m = 0$: There exist a shortest path from v_i to v_j with no edges $\leftrightarrow$ $i = j$.

$$\blacktriangleright\ d_{ij}^{0} = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{if } i \ne j \end{cases}$$

- $m \ge 1$:

$$d_{ij}^{m} = \min\{\, d_{ij}^{m-1},\ \min_{1 \le k \le n \wedge k \ne j} \{ d_{ik}^{m-1} + \omega_{kj} \}\}$$

$= \min_{1 \le k \le n} \{ d_{ik}^{m-1} + \omega_{kj} \}$ for all $v_k \in V$, since $\omega_{jj} = 0$ for all $v_j \in V$.

Shortest Paths

(3) Computing the shortest-path weights bottom-up :

- given $W = D^1$, compute a series of matrices $D^2, D^3, \ldots, D^{n-1}$,

  where $D^m = (\, d_{ij}^{m} \,)$ for $m = 1, 2, \ldots, n-1$

  - ▶ final matrix D^{n-1} contains actual shortest path weights,

    i.e., $d_{ij}^{n-1} = \delta(v_i, v_j)$

- SLOW-APSP(W)

```
D^1 ← W
for m ← 2 to n-1 do
    D^m ← EXTEND( D^(m-1) , W )
return D^(n-1)
```

8

Shortest Paths

```
EXTEND ( D , W )
▶ D = ( d_ij ) is an n x n matrix
  for i ← 1 to n do
     for j ← 1 to n do
        d_ij ← ∞
        for k ← 1 to n do
           d_ij ← min{d_ij , d_ik + ω_kj}
  return D
```

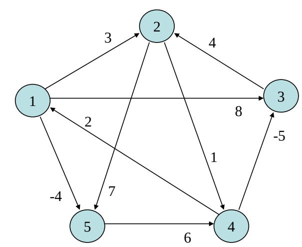
Shortest Paths

- relation to matrix multiplication C = A × B : $\mathbf{c}_{ij} = \sum_{1\le k\le n} \mathbf{a}_{ik} \times \mathbf{b}_{k\,j}$,
  ▶ $D^{m-1} \leftrightarrow A$ & $\mathbf{W} \leftrightarrow B$ & $D^{m} \leftrightarrow C$

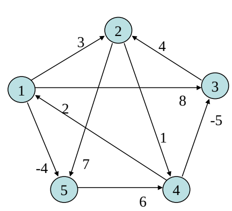
- Thus, we compute the sequence of matrix products
  - $D^1 = D^0 \times W = W$; note D^0 = identity matrix, i.e., $d_{ij}^{0} = \begin{cases} 0 & \text{if } i = j \\ \infty & \text{if } i \neq j \end{cases}$
  - $D^2 = D^1 \times W = W^2$
  - $D^3 = D^2 \times W = W^3$
  - •
  - $D^{n-1} = D^{n-2} \times W = W^{n-1}$

Shortest Paths

- Example



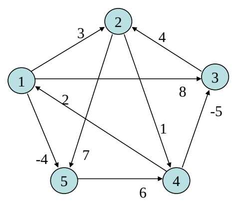
Shortest Paths



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

$D^1 = D^0 W$

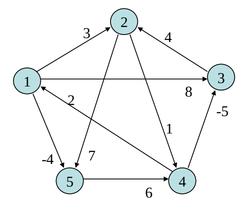
Shortest Paths



	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

$D^2 = D^1 W$

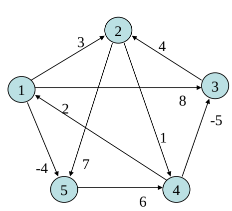
Shortest Paths



	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



	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

$D^4 = D^3 W$