All Pair Shortest Path

1. Bellman-Ford n times $= O(n \cdot nm) = O(n^2 m) = O(n^4)$

   ($\hookrightarrow$ adj. list representation) for dense.

2. Dijkstra $O(n \cdot n^2) = O(n^3)$ for dense

   ($\hookrightarrow$ priority queue of edges)

   $$O(n \, (n+m) \log n) = O((n^2 + mn) \log n)$$
   $$= O(n^2 \log n) \text{ for sparse}$$

- We will do 2 dynamic prog. algorithms with _tight_ codes (small coefficients) on $O(n^3 \log n)$ and another $O(n^3)$.

goal create $n \times n$ matrix of shortest-path distances $\delta(u,v)$,

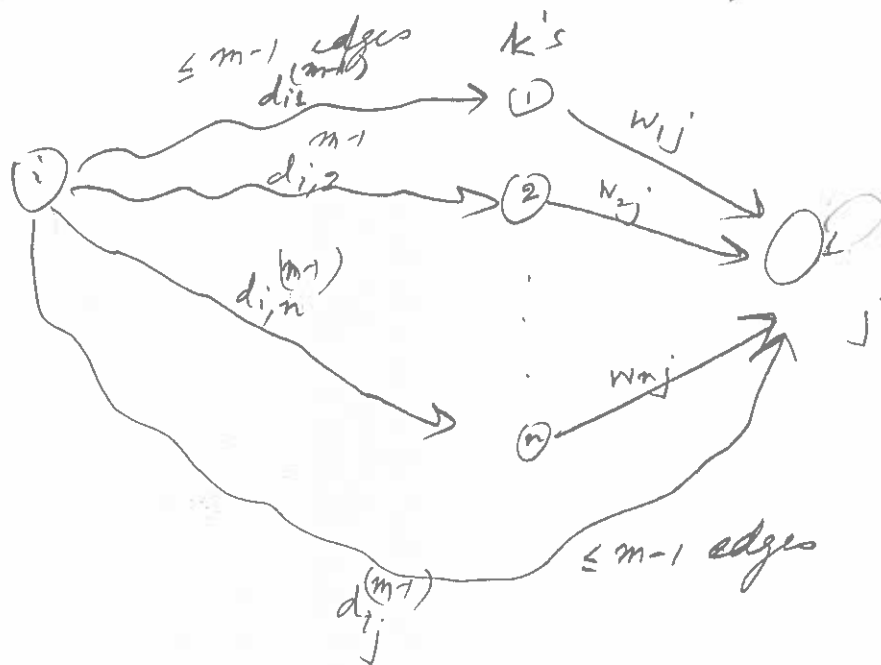graph Representation (adj. matrix)

$$W_{n \times n} = (w_{ij}) \text{ of edge weights.}$$

$$w_{ii} = 0$$

$\uparrow$ (m) $\boxed{\# |E|}$

Let $d_{ij}^{(m)} =$ weight of shortest path from i to j that uses _at most_ m edges

$$d_{ij}^{(0)} = \begin{cases} 0 & i = j \\ \infty & i \neq j \end{cases}$$

$$d_{ij}^{(m)} =$$

Top figure labels: $\le m-1$ edges, k's, $d_{ik}^{(m-1)}$, w_{1j}, $d_{i,2}^{m-1}$, w_{2j}, $d_{i,n}^{(m-1)}$, w_{nj}, node j, $\le m-1$ edges, $d_{ij}^{(m-1)}$

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

$$= \min\left\{ d_{ij}^{m-1}, \left\{ d_{ij}^{(m-1)} + w_{jj} \right\}_{k=j}, \min_{k \ne j}\{ d_{ik}^{(m-1)} + w_{nj}\} \right\}$$

$$d_{ij}^{(m)} = \min_{1 \le k \le n}\{ d_{ik}^{(m-1)} + w_{kj}\} \qquad —— \textcircled{1}$$

<u>Answer</u>: $\delta(i,j) = d_{ij}^{(m-1)} = d_{ij}^{(n)} = d_{ij}^{(n+1)} = \cdots$

Pseudocode for 'relaxation step'

```
for  k ← 1 to n                              O(n)
      if  dij > dik + wkj
          then  dij ← dik + wkj
```

$d_{ij}^{(m)}$, $O(n^2 \cdot n)$

Time $= O\left(n \cdot n^3 \right) = O(n^4)$

Similarity to Matrix Multiplication

$$C_{n\times n} = A_{n\times n} \cdot B_{n\times n} \quad n\times n \quad \text{matrices}$$

$$c_{ij} = \sum_{1\le k\le n} a_{ik} \cdot b_{kj} \qquad - ②$$

Compare with

$$d_{ij}^{(m)} = \min_{1\le k\le n} \left\{ d_{ik}^{(m-1)} + w_{kj} \right\} \qquad ①$$

Replace
$$c_{ij} \to d_{ij}^{(m)}$$
$$a_{ik} \to d_{ik}^{(m-1)}$$

$$b_{kj} \to w_{kj}$$
$$+ \; \to \; \min$$
$$\cdot \; \to \; +$$

$$\Rightarrow \quad D^{(m)} = D^{(m-1)} \text{"}X\text{"} \; W \qquad \text{———} \; ③$$

identity g "X"

met
mult

$$\begin{bmatrix} 1 & 0 & 0 & \cdots & 0 \\ 0 & 1 & & & \\ 0 & & \ddots & & \\ 0 & & & & 1 \end{bmatrix} \to \begin{bmatrix} 0 & \infty & \infty & \cdots & \infty \\ & 0 & & & \\ & & \ddots & & \\ & & & & 0 \end{bmatrix}$$

$$= D^{(0)}$$

$$\boxed{\begin{array}{cc} \min & + \\ \infty \leftarrow 0 & \end{array}}$$

$$\boxed{\begin{array}{c} + \leftarrow * \\ 0 \leftarrow 1 \end{array}}$$

$$D^{(1)} = D^0 W = W$$
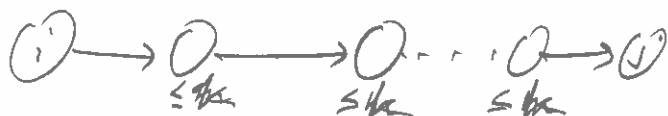$$D^2 = D^1 W = W^2$$
$$D^{(n-1)} = W^{n-1}$$

$$n\left(\Theta(n^3)\right) = \Theta(n^4?)$$

But $\quad W \to W^2 \to W^4 \to \cdots \to W^{2\lceil \log_2 n-1\rceil}$

overshoot, no problem
$\Theta(n^3 \log n)$

Floyd-Warshall Algorithm

Dynamic Prog but faster : $\Theta(n^3)$

$c_{ij}^{(k)}$ = weight of a shortest path from i to j
with intermediate vertices in $\{1, 2, \ldots, k\}$

$$i \rightarrow \bigcirc \rightarrow \bigcirc \cdots \bigcirc \rightarrow j$$
$$\leq k \qquad \leq k \qquad \leq k$$

Then $\quad \delta(i,j) = c_{ij}^{n}$

$$c_{ij}^{0} = w_{ij}$$

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

$$\begin{array}{c} k \\ \overset{(k-1)}{c_{ik}} \nearrow \bigcirc \searrow \overset{(k-1)}{c_{kj}} \\ i \bigcirc \longrightarrow \longrightarrow j \\ \overset{(k-1)}{c_{ij}} \end{array}$$

intermediates in $\{1, 2, \ldots, k-1\}$

pseudocode

```
for k ← 1 to n
    for i ← 1 to n
        for j ← 1 to n
            c_ij^k = min ( c_ij^{k-1} , c_ik^{k-1} + c_kj^{k-1} )
```

$\Theta(n^3)$.

space $O(n^2)$

ALL-TO-ALL SHORTEST PATHS

$$D^k[i,j] = \text{the weight of a shortest path for } v_i \text{ to } v_j \text{ using nodes from } \{v_1, v_2, \ldots v_k\} \text{ as intermediate vertices in the path.}$$

$$D^{(0)} = (w_{ij})_{n \times n} = \begin{cases} 0 & i = j \\ \infty & \{i,j\} \notin E \\ \text{weight of } \{i,j\} \end{cases}$$

$$D^{(m)} = \text{output}$$

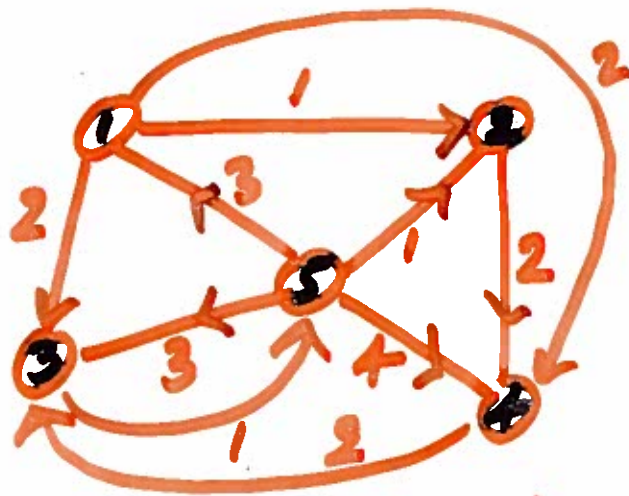v_i 〇 —— $D^{k-1}[i,j]$ ——> 〇 v_j

$D^{k-1}[i,k]$ ——> 〇 v_k —— $D^{k-1}[k,j]$
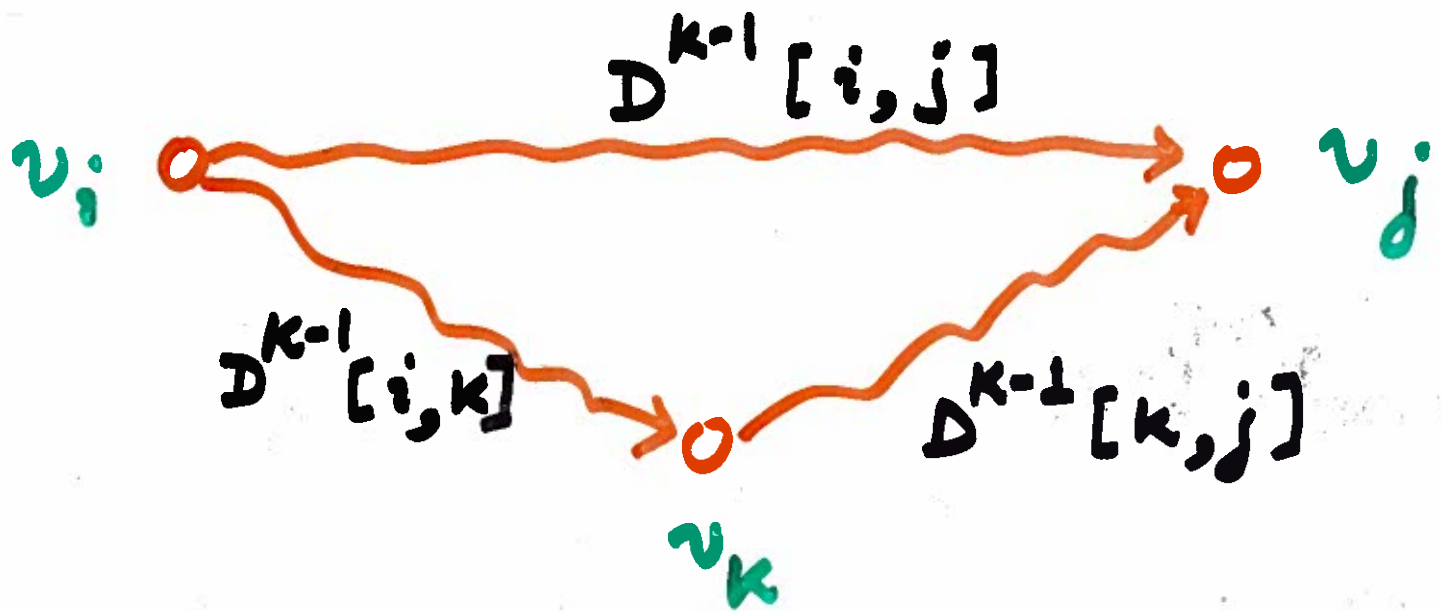
$$D^{(k)}[i,j] =$$

$$\min\left[D^{(k-1)}[i,j], \; D^{(k-1)}[i,k] + D^{(k-1)}[k,j]\right]$$

ALL-TO ALL

D = Weight Matrix

for $k \leftarrow 1$ to n do

    for $i \leftarrow 1$ to n do

    for $j \leftarrow 1$ to n do

$$D[i,j] = \min \left[D[i,j], \; D[i,k] + D[k,j] \right]$$

endfor ; endfor

endfor.

$$D^{(0)} =$$

	1	2	3	4	5
1	0	1	2	2	∞
2	∞	0	∞	2	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	∞
5	3	1	3	4	0

	1	2	3	4	5
1	0	1	2	2	∞
2	∞	0	∞	2	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	∞
5	3	1	3	4	0

$$D^{(1)}$$

	1	2	3	4	5
1	0	1	2	2	∞
2	∞	0	∞	2	∞
3	∞	∞	0	∞	1
4	∞	∞	2	0	∞
5	3	1	3	3	0

$$D^{(2)}$$

(2.1)

$D^{(3)}$

0	1	2	2	3
∞	0	∞	2	∞
∞	∞	0	∞	1
∞	∞	2	0	3
3	1	3	3	0

$D^{(4)}$

0	1	2	2	3
∞	0	4	2	5
∞	∞	0	∞	1
∞	∞	2	0	3
3	1	3	3	0

$D^{(5)}$

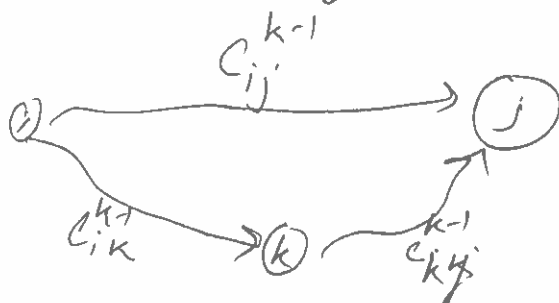|				
0	1	2	2	3
8	0	4	2	5
4	2	0	4	1
6	4	2	0	3
3	1	3	3	0

(9.2)

Transitive Closure

$$G^* = (V, E^*)$$

$(i,j) \in E^*$ iff $\exists$ path from i to j in G.

Solution

Adj Mat $(0,1)$ matrices

In Floyd Warshol

$$c_{ij}^k = c_{ij}^{k-1} \text{ OR } \left(c_{ik}^{k-1} \text{ AND } c_{kj}^{k-1} \right)$$

$$O(n^3)$$