------------------------------ 25 ------------------------------ 24-3a

All-Pairs Shortest Paths

Input: the adjacent matrix *W* of a weighted directed graph *G*=(*V*, *E*), where

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

(Negative weights can present. But, *G* contains ☆ no negative-weight cycles.)

simple path, at most n – 1 edges

$$\overbrace{i \rightsquigarrow \pi_{ij}}^{d_{ij}} \rightarrow j$$

Output: A matrix $D=(d_{ij})$, where $d_{ij} = \delta(i, j)$

A *predecessor matrix* $\Pi=(\pi_{ij})$, where π_{ij} is the predecessor of *j* on some shortest path from *i*.

(Subgraph induced by row *i* of Π is a shortest-paths tree with root *i*.) single source

25-1x

25.1 Shortest paths and multiplication

(A dynamic-programming approach)

Optimal structure: all subpaths of a shortest path are shortest paths.

A recursive solution:

Let $d_{ij}^{(m)}$ be the minimum weight of any path from *i* to *j* that contains at most *m* edges. 25-2a

D^0 (boundary cond.)

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

(or $D^1 = W$)

D^m

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

$D^m = D^{m-1} \otimes W \quad (op_1, op_2) = (+, \min)$ 25-2c

Since *G* contains no negative-weight cycles,

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

a simple path has at most n – 1 edges

* $D^{(1)} = W$

* $D = D^{(n-1)} = D^{(n)} = D^{(n+1)} = \ldots$

(See 25-4 Fig)

Computing $D^{(m)}$ from $D^{(m-1)}$

D^{m-1}

EXTEND-SHORTEST-PATHS(D, W)

$d'_{ij} = \mathrm{MIN}_k\{d_{ik} + w_{kj}\}$

$D' = D \otimes W$

$O(n^3)$

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

* D for $D^{(m-1)}$ and D' for $D^{(m)}$
* Time: $\Theta(n^3)$
* Similar to matrix multiplication $C=A\times B$:

$d_{ij}^{(m-1)}$ --> a_{ij} ω_{ij} --> b_{ij} $d_{ij}^{(m)}$ --> c_{ij}

min --> + (Σ) + --> *

MATRIX-MULTIPLY(A, B)

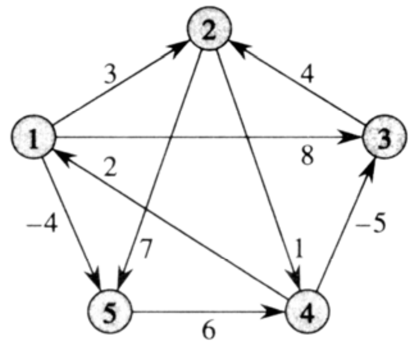
$C = A \times B$

$O(n^3)$

$n \leftarrow rows[A]$
let C be an $n \times n$ matrix
for $i \leftarrow 1$ **to** n
  do for $j \leftarrow 1$ **to** n
    do $c_{ij} \leftarrow 0$
      for $k \leftarrow 1$ **to** n
        do $c_{ij} \leftarrow c_{ij} + a_{ik} \cdot b_{kj}$
return C

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

$D^{(3)} = D^{(2)}W = W^3$ ... $D^{(m)} = W^m$



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

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

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

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

SLOW-ALL-PAIRS-SHORTEST-PATHS(W)

$n \leftarrow rows[W]$
$D^{(1)} \leftarrow W$
for $m \leftarrow 2$ **to** $n-1$
  do $D^{(m)} \leftarrow$ EXTEND-SHORTEST-PATHS($D^{(m-1)}$, W)
return $D^{(n-1)}$

$(n-2)*O(n^3)$

$W = D_1 \xrightarrow{*W} D_2 \xrightarrow{*W} D_3 \cdots \xrightarrow{*W} D^{n-1}$

* Time: $(n-2) \times O(n^3) = O(n^4)$.

* Space: $O(n^2)$
(Note that only two matrix is really required.)

Improving the running time by repeated squaring

$$W^2 = W \times W \qquad W^4 = W^2 \times W^2$$

$$W^8 = W^4 \times W^4 \qquad \ldots$$

$$W^{2^{\lceil \lg(n-1) \rceil}} = W^{2^{\lceil \lg(n-1) \rceil - 1}} \times W^{2^{\lceil \lg(n-1) \rceil - 1}} = D$$

$m = \lceil \lg (n-1) \rceil$ times ($2^m \geq n-1 \Rightarrow m \geq \lceil \lg (n-1) \rceil$)

(Note that $D = W^{n-1} = W^n = W^{n+1} = \ldots$)

FASTER-ALL-PAIRS-SHORTEST-PATHS(W)

```
n ← rows[W]
D^(1) ← W
m ← 1
while n − 1 > m
      do D^(2m) ← EXTEND-SHORTEST-PATHS(D^(m), D^(m))
         m ← 2m
return D^(m)
```

lg n * $O(n^3)$

$D^1 \xrightarrow{*D^1} D^2 \xrightarrow{*D^2} D^4 \xrightarrow{*D^4} D^8 \xrightarrow{*D^8} \cdots D^m$

$m \geq n - 1$
(no negative cycles)

Time: $\Theta(n^3 \lg n)$

Note: also can also check negative cycles

25.2 The Floyd-Warshall algorithm

(A dynamic-programming approach)

A recursive solution:

Let $d_{ij}^{(k)}$ be the weight of a shortest path from i to j with all ***intermediate*** vertices in $\{1, 2, \ldots, k\}$. 25-6a

$$d_{ij}^{(0)} = \omega_{ij} \quad (D^{(0)} = W)$$

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

(since G contains no negative-weight cycles)

* $d_{ij} = \delta(i, j) = d_{ij}^{(n)}$. $(D = D^{(n)})$ visit a vertex at most once (simple path)

FLOYD-WARSHALL(W)

```
n ← rows[W]
D^(0) ← W
for k ← 1 to n
     do for i ← 1 to n
          do for j ← 1 to n
               do d_ij^(k) ← min(d_ij^(k−1), d_ik^(k−1) + d_kj^(k−1))
return D^(n)
```

$D^{(0)} \rightarrow D^{(1)} \rightarrow D^{(2)} \rightarrow \cdots \rightarrow D^{(n)}$

Time: $\Theta(n^3)$

25-6b

Note: recurrence is incorrect if there are negative cycles

Constructing a shortest path: Refer to textbook π

Transitive closure of a directed graph G = (V, E)

25-7a

$G^* = (V, E^*)$, $A \longrightarrow A^*$
adjacency matrix → transitive closure

where $E^* = \{(i, j) \mid$ if there is a path from i to j in $G\}$.

25-7b

Method 1: assign a weight 1 to each edge of G and then perform Floyd-Warshall algorithm. We have (i, j) in E^* iff $d_{ij} < n$.

$O(n^3)$ time
$O(n^2)$ space

25-7c

Method 2: (Save time and space in practice)

Define $t_{ij}^{(k)}$=1 if there is a path from i to j with all ***intermediate*** vertices in $\{1, 2, ..., k\}$.

$$t_{ij}^{(0)} = \begin{cases} 0 & \text{if } i \neq j \text{ and } (i,j) \notin E \\ 1 & \text{if } i = j \text{ or } (i,j) \in E \end{cases} \qquad T^{(0)}$$

☆

$$t_{ij}^{(k)} = \underbrace{t_{ij}^{(k-1)}}_{\text{不必經 k 就可走到}} \vee \underbrace{(t_{ik}^{(k-1)} \wedge t_{kj}^{(k-1)})}_{\text{先到 k, 再走過來}} \qquad T^{(k)}$$

或

$A^* = T^{(n)}$

TRANSITIVE-CLOSURE(G)

```
n ← |V[G]|
for i ← 1 to n
    do for j ← 1 to n
           do if i = j or (i, j) ∈ E[G]
                 then t_ij^(0) ← 1
                 else t_ij^(0) ← 0
for k ← 1 to n
    do for i ← 1 to n
           do for j ← 1 to n
                  do t_ij^(k) ← t_ij^(k-1) ∨ (t_ik^(k-1) ∧ t_kj^(k-1))
return T^(n)
```

$T^{(0)}$ (lines 2–6), $T^{(k)}$ (lines 7–10), $T^{(n)} = A^*$

$T^{(0)} \rightarrow T^{(1)} \rightarrow T^{(2)} \rightarrow \cdots \rightarrow T^{(n)}$

* **Time:** $\Theta(n^3)$ boolean operations

* Only 1 bit is required for each $t_{ij}^{(k)}$.
Space: $O(n^2)$ bits

* G^* can be used to determine the strongly connected components of G.

25-8a

Homework: Ex. 25.1-5, 25.1-6, 25.1-10, 25.2-3, 25.2-4, 25.2-8, Pro. 25-1.

↳ 簡化版 Floyd-Warshall (請自己欣賞!)