Recursive Solution

$l_{ij}^{(m)}$ m개의 edge까지 고려했을때 i-j 까지의 최단거리의 weight 값

$m=0 : l_{ij}^{(0)} = \begin{cases} 0 & \text{if } i=j \\ \infty & \text{if } i \neq j \end{cases}$ $\quad m \geq 1 : l_{ij}^{(m)} = \min\left(l_{ij}^{(m-1)}, \min\limits_{1 \leq k \leq n} \{ l_{ik}^{(m-1)} + w_{kj} \} \right)$

어차피 여기에 포함

$$= \min_{1 \leq k \leq n} \{ l_{ik}^{(m-1)} + w_{kj} \}$$

m=1 일때 $l_{ij}^{(1)} = w_{ij}$

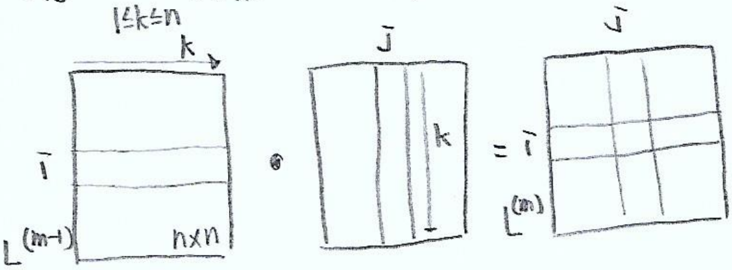
$L^{(1)} = W$ (엣지들의 weight 배열)

$L^{(m)} = \left(l_{ij}^{(m)} \right)$

$L^{(m-1)}$과 W로 $L^{(m)}$을 계산할 수있다.

negative cycle 이 없는 그래프에서는
모든 shortest path 의 엣지수는 n-1 개 이하이다.

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



$L^{(m)}$을 계산하는것은 마치 행렬곱 같다.

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

```
Extend(L,W,n)
Create L', an nxn matrix
for i <- 1 to n
    do for j <- 1 to n
        do l'ij <- ∞        일단 ∞로 초기화
            for k <- 1 to n
                do l'ij <- min( l'ij , lik + wkj )
return L'
```

L와 W로 L'을 만드는과정

$O(n^3)$

SLOW-APSP (W,n) — All pairs shortest paths

```
L(1) <- W
for m <- 2 to n-1
    do L(m) <- Extend ( L(m-1), W, n )
return L(n-1)
```

$O(n^4)$

Floyd Warshall (노드를 이용)

Input. Directed, weighted graph $G=(V,E)$
음수 edge 가능.

output. The shortest paths between all pairs of vertices in a graph

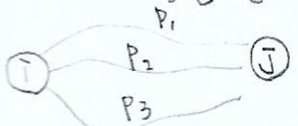
• 그래프의 vertex는 $V=\{1,2,\cdots n\}$ 으로 주어진다.

Path가 $p=\langle v_1, v_2, v_3 \cdots v_j \rangle$ 로 주어지면
중간 vertex는 $\{v_2, v_3 \cdots v_{j-1}\}$ Set으로 나타난다.

$p=\langle 1,2,4,5\rangle : \{2,4\}$

$p=\langle 2,4,5\rangle : \{4\}$

그래서 $i, j \in V$의 경로를 구할때는 $\{1,2,\cdots k\}$ Set의 모든 subset을 고려한다.



이경로의 vertex들은 k를 넘지 않는다.

~~k는 Set의 원소의 수를 의미하지 않음~~

$d_{ij}^{(k)}$ i에서 j로가는 shortest path의 weight k는 $\{1,2,\ldots k\}$ Set~~의 원소의 수~~

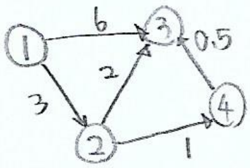
$d_{13}^{(0)}=6 \quad \{\varnothing\}$

$d_{13}^{(1)}=6 \quad \{1\}$

$d_{13}^{(2)}=5 \quad \{1,2\}$

$d_{13}^{(3)}=5 \quad \{1,2,3\}$

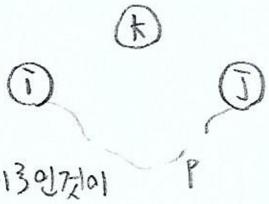
$d_{13}^{(4)}=4.5 \quad \{1,2,3,4\}$



~~안에 있는건 "써야됨"~~
써도되고, 안써도됨

• k가 path p에 포함되지 않을때

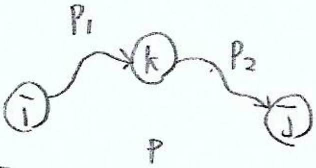
i→j로가는 shortest path는 중간 vertex가 from $\{1,2,\ldots k\}$ 인것과 from $\{1,2,\ldots k-1\}$ 인것이 같다. (어차피 k가 들어가도 shortest path에는 포함이 안돼서)



• k가 p의 중간 vertex일때

k는 P_1, P_2의 중간 vertex가 아니다.

• P_1 is a shortest path from i to k with vertex from $\{1,2,3,\cdots k-1\}$
• P_2 is a shortest path from k to j with vertex from $\{1,2,3,\cdots k-1\}$



$d_{ij}^{(k)}$는 the weight of a shortest path from vertex i to j with all intermediary vertics drawn from $\{1,2,\ldots k\}$

$k=0 \rightarrow d_{ij}^{(k)} = w_{ij}$

$k \geq 1$

Case 1: k is not an intermediate vertex of path p

$$d_{ij}^{(k)} = d_{ij}^{(k-1)}$$

Case 2: k is an intermediate vertex of path p

$$d_{ij}^{(k)} = d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$$

그래서

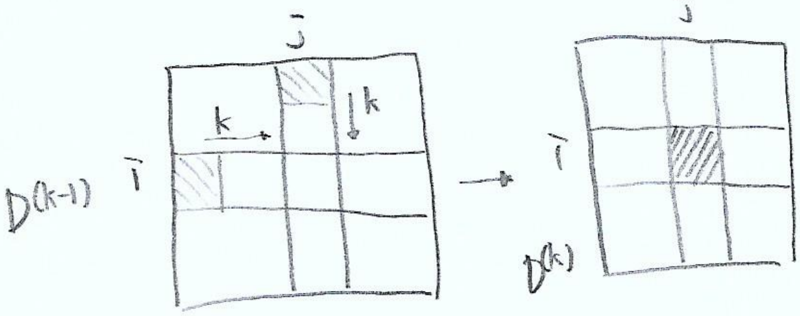
$$d_{ij}^{(k)} \begin{cases} w_{ij} & k=0 \\ \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\} & k \geq 1 \end{cases}$$

The Final solution

$D^{(n)} = (d_{ij}^{(n)}) : \delta(i,j) \ \forall\, i,j \in V$

$\delta(i,j)$: all simple shortest paths contain at most $n-1$ edges

$\delta_{ij}^{(1)}, \delta_{ij}^{(2)} \cdots \delta_{ij}^{(n-1)}$



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)
```

$O(n^3)$

π 경로를 나타냄

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

$$\pi_{ij}^{(k)} = \begin{cases} \pi_{ij}^{(k-1)} & d_{ij}^{(k-1)} \leq d_{ik}^{(k-1)} + d_{kj}^{(k-1)} \\ \pi_{kj}^{(k-1)} & \text{otherwise} \end{cases}$$ 경로가 안 바뀔때

PPT에서 배열 채우는거 꼭 해보기!