ALL to ALL shortest path Problem

-이전의 알고리즘은 한 출발점에서 최단 경로를 구함.

- All to All은 모든 정점에서의 최단 경로를 구함.

더 정확한 표현

- 방법1: for each node s in V:
  Dijksra (s) : $O(t \log n)$ } $O(nm \log n)$

최대 n^2까지

$= O(n^3 \log n)$

- 방법2 DP 방: Floyd-warshall 알고리즘.

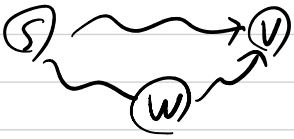
노드 = $\{1, 2, 3, 4 \cdots n\}$

```
for i,j ∈ {1, 2, ⋯ n}          n²
    P(i,j) 계산                # p(i,j) = 노드 i에서 노드 j까지의 최단 경로
    d(i,j) 계산 O(n)           # d(i,j) = |P(i,j)| (최단경로 길이)
    ↳ DP 계산
```

$O(n^3)$

- Relax(w,v): w → v

```
if dist[w] + cost(w,v) < dist[v]
    dist[v] = dist[w] + cost(w,v)
    parent[v] = w
```



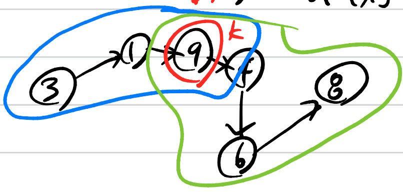
노드 집합 = $\{1, 2, 3, \cdots n\}$

1. 해를 분석

최단경로

$|SP(i,j)| = |SP(i,k)| + |SP(k,j)|$

DP식 = $d(i,j) = d(i,k) + d(k,j)$



K는 인덱스 번호가 가장 큰 놈으로 설정.

$P_9(3,8) = P_8(3,9) + P_8(9,8)$

↳ 3→8 경로 중에서

K=9를 거쳐 가는게 최단 경로이다.

$P_n(3,8) = P_{n-1}(3,n) + P_{n-1}(n,8)$

$P_k(i,j) = \text{노드 } i \underset{\le k}{\rightsquigarrow} \text{노드 } j = \overbrace{\underset{\le k-1}{P_{k-1}(i,k)} + \underset{\le k-1}{P_{k-1}(k,j)}}^{\le k}$

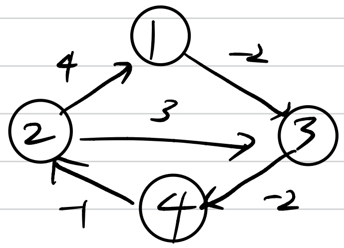
$d_0 = \begin{cases} w(i,j) \\ \infty \end{cases}$

$d_0(i,i) = 0$

DP식

$d(i,j) = \min(d_{(i,j)}, d_{k-1}(i,k) + d_{k-1}(k,j))$

d	1	2	3	4 ⋯	n
1					
2					
3					
4					
⋮					
n					



d_4
~~d_3~~
~~d_2~~
~~d_1~~

~~d_0~~	1	2	3	4
1	0	∞	-2	∞
2	4	0	3	∞
3	∞	∞	0	-2
4	∞	-1	∞	0

$$d_k(i,j) = \min \begin{cases} d_k(i,j) \\ d_{k-1}(i,k) + d_{k-1}(k,j) \end{cases}$$

- K=0 → K=1

$$d_1(i,j) \underset{\min}{=} \begin{cases} d_0(i,j) \\ d_0(i,1) + d_0(1,j) \end{cases}$$

ex)

$$d_1(2,3) = \min\left(\underset{=3}{d_0(2,3)},\ \underset{=2}{\overset{4\ +\ -2}{d_0(2,1) + d(1,3)}} \right)$$

$$= 2$$

1. d: 2차원 리스트 n×n 초기값 ∞ (inf)

2. d초기

$$d[i][j] = \begin{cases} w(i,j) \\ \infty \end{cases}$$

$$d[i][i] = 0$$

```
3. for k = 1 to n        n  > O(n^3)
       for (i,j) :  O(n^2)
           $d[i][j] = \min(d_{k-1}[i][j],\ d_{k-1}[i][k] + d_{k-1}[k][j])$
```