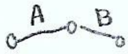
최단거리 (MST랑 헷갈릴수 있음)

특정노드에서 노드까지의 최단경로

if path p is a shortest path → p의 subpath is a shortest path

증명

Path p is a shortest path ∩ p의 subpath is not a shortest path



shortest path P

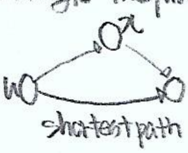
A가 shortest path가 아닐경우 shortest path A' 존재

$A + B = P$, V   p가아닌 p'가 shortest path이므로   path p is a shortest path (False)

$A' + B = P'$ 역이 False 이므로 명제는 true

triangle inequality

u→v 가 shortest path라면

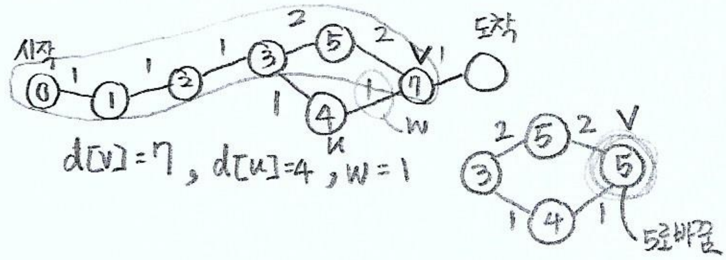


$w(u \to v) \le w(u \to x) + w(x \to v)$

relaxation

Relaxation (u, v, w)

if (d[v] > d[u]+w) then d[v] = d[u]+w



Bellman Ford (edge를 이용함)

```
for each v ∈ V
    d[v] = ∞
d[s] = 0
for i = 1 to |V|-1
    for each edge (u,v) ∈ E
        Relax(u, v, w(u,v))
```

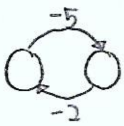
V-1 번만큼 한번 할때마다 모든노드를 Relax()

(V-1) × E

O(V·E)

```
for each edge (u,v) ∈ E
    if (d[v] > d[u] + w(u,v))
        return "no solution"
```

검증. ↑ 위에서 다 끝났는데 다시 update가 일어난다는건? negative cycle이 발생한다는것.



← Relax 할때마다 d[]가 줄어든다. 끝이나지 않아서 "no solution"

Dijkstra's Algorithm

Prim이랑 비슷하지만 값을 누적해서 넣는것. prim에서 두줄만 바꾸면 된다.

```
Q = V[G]
for each u∈Q
    key[u] = ∞
key[r] = 0
P[r] = NULL
while (Q not Empty)
    u = ExtractMin(Q)
    for each v∈Adj[u]
        if (v∈Q and w(u,v)+key[u] < key[v])
            P[v] = u
            key[v] = w(u,v) + key[u]
```

Q의 구조에 따른 시간의 차이 표도 같다.

지금까지는 하나의 노드에 대한경로. 이것을 전체노드에 대해 구하면 (All pairs shortest paths)

Bellman-Ford → $V \times O(V \cdot E) = O(V^2 \cdot E)$, 그래프가 Fully Conneted 이면 $(E = O(V^2)) \rightarrow O(V^4)$

Dijkstra's → binary heap 일때 → $O(V \cdot E\lg V)$ → Fully Connected $O(V^3 \lg V)$

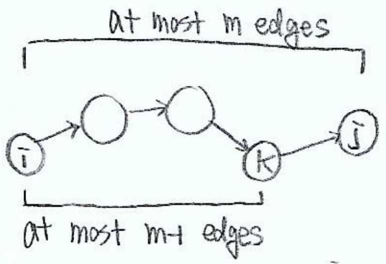
Fibonacci heap일때 → $O(EV + V^2 \lg V)$ → Fully Connected $O(V^3)$

이제 모든경우에 $O(V^3)$ 인 알고리즘을 볼것이다.

W = V×V개의 2차원배열 (Edge의 Weight가 기록되어있다)

D = Output. 최단거리가 담긴 2차원배열.

Optimal substructure of a shortest path.



i→j까지의 shortest path를 p라하자.

If $i = j$

$w(p) = 0$, p 는 edge가 없다.

If $i \neq j$

$p = i \overset{p'}{\rightsquigarrow} k \rightarrow j$

- p'은 (edge가 1개 … m-1개로 구성된) 경로중 shortest path.

$\delta(i, j) = \delta(i, k) + w_{kj}$ (δ은 w을 의미. (엣지들의 Weight의 합))