

All-Pairs Shortest Path

모든 쌍의 최단거리 찾기 \rightarrow 답이 n^2 개

Dijkstra n 번 하면 됨. $O(n \times (n+m) \log n) \rightarrow$ 제일 클 때 $O(n^3 \log n)$

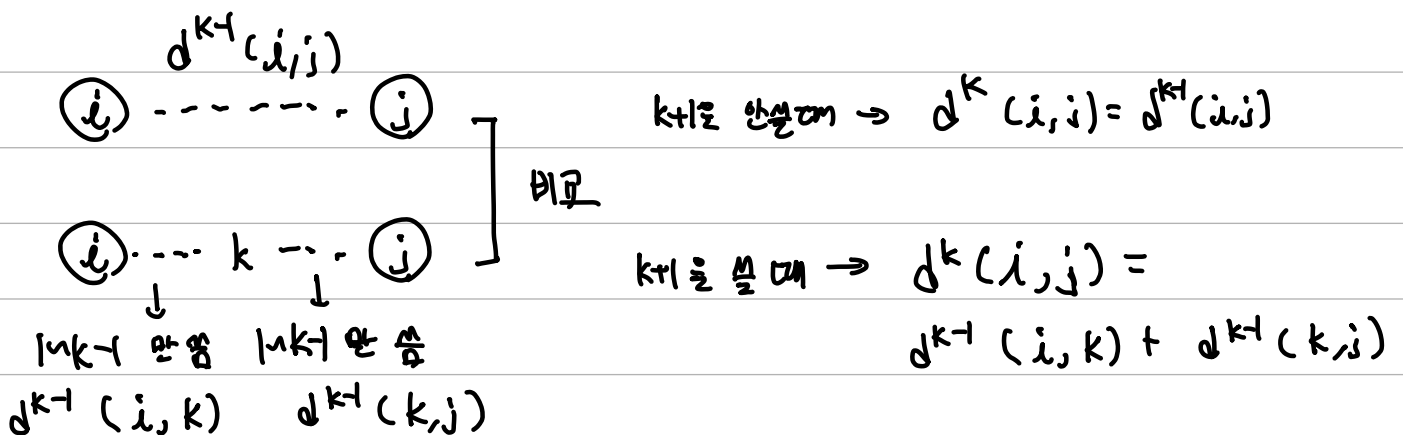
\downarrow $O(n^3)$ 자리 알고리즘?

Floyd Warshall

$d^k(i, j)$: 모든 i 에서 j 까지의 shortest path 길이 ($1 \sim k$ 노드만 거쳐서)

$k=0$: i 부터 j 까지 바로 가는 길

$k-1$ $d^{k-1}(i, j)$ 지
 \downarrow
 k $d^k(i, j)$ 구하기



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

d^k 를 구할 때 d^{k-1} 의 값만 사용하므로 \rightarrow 메모리 축소 가능!

· 2차원 배열 2개를 k 가 늘, 작아져가면서 쓰면 됨.

· 2차원 배열 1개만 쓸 땐,

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

or $\begin{cases} d^{k-1}(i, k) \\ d^k(i, k) \end{cases}$ 이쪽만 써도 됨.

왜? i 에서 k 로 가는 데 $i \rightarrow k$ 를 쓰거나 $i \rightarrow k$ 를 쓰거나 같음.

k 를 거치나 안거치나 k 가 도착점 이어서 Shortest Path에는 변화 X