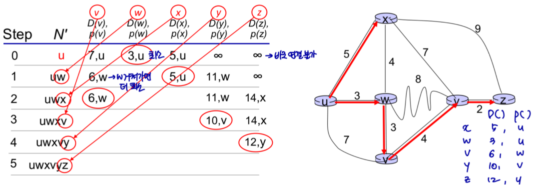
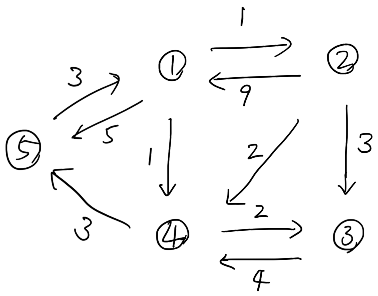


6.6.2 All-pairs shortest path

꼭짓점이 5개니까
edge 4번이 최대"

Now, consider the following graph and its distance matrix:

<weighted>



$$D = \begin{array}{c} \\ 1\\ 2\\ 3\\ 4\\ 5 \end{array}\begin{array}{c} \begin{array}{ccccc} 1 & 2 & 3 & 4 & 5 \end{array} \\ \begin{pmatrix} 0 & 1 & \infty & 1 & 5 \\ 9 & 0 & 3 & 2 & \infty \\ \infty & \infty & 0 & 4 & \infty \\ \infty & \infty & 2 & 0 & 3 \\ 3 & \infty & \infty & \infty & 0 \end{pmatrix} \end{array}$$

Step	2	3	4	5
0 ①	1	∞	1	5
1 ①②	X	4	①	5
2 ①②④	X	③	X	4
3 ①②④③	X	X	X	④
4 ①②④③⑤				

In general, we consider a graph G whose vertices are $1, 2, \ldots, n$, and will consider a sequence of matrices

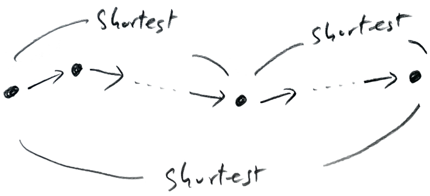
$$D = \underbrace{D^{(0)}}_{\text{직접연결}}, \overset{\text{1개 중간점}}{D^{(1)}}, \overset{\text{2개 "}}{D^{(2)}}, \ldots, \boxed{D^{(n)}}, \text{ 최종 최단거리}$$

i시작점, j끝점

where $D^{(k)}$ is the k-distance matrix, that is, its (i, j)th entry $D^{(k)}_{i,j}$ is the length of the shortest path from vertex i to j, *using only vertices* $1, 2, \ldots, k$. (In short, k-distance between i and j.) Make sure that $D^{(0)} = D$ makes sense, since $D_{i,j}$ is the distance between i and j without using any other vertices as intermediates, i.e., only the edge (i, j). The matrix $D^{(n)}$ contains the information that we are seeking, i.e., the shortest distances between two vertices.

Recursive structure of optimality Note that we are using $1, 2, \ldots, n$ as vertices. In the following, v_i's are variables that take a value from $1, \ldots, n$.

Proposition 17. *If $v_1 v_2 \ldots v_m \ldots v_l$ be a shortest path from v_1 to v_l, then the subpaths $v_1 \ldots v_m$ and $v_m \ldots v_l$ are shortest paths. Also, the same thing holds for paths that uses only $1, \ldots, k$ as intermediates.*



Proof. Suppose that $v_1 v_2 \ldots v_{m-1} v_m$ is not shortest and there is a shorter path from $v_1 v'_2 \ldots v'_{m-1} v_m$. Then there is a shorter path $v_1 v'_2 \ldots v'_{m-1} v_m \ldots v_l$, a contradiction. Therefore, $v_1 v_2 \ldots v_{m-1} v_m$ is the shortest, and the same argument holds for $v_m \ldots v_l$. Also, the claim about paths using only $1, \ldots, k$ as intermediates is proved similarly. □

By definition of $D^{(k)}$ and the above observations, the following holds.

Proposition 18.

$$D^{(k)}_{i,j} = \min\left(\overset{\text{k사용X최단}}{D^{(k-1)}_{i,j}} \,/\, \overset{\text{k중간점으로 설정한 최단}}{\underbrace{D^{(k-1)}_{i,k}}_{\text{i~k까지}} + \underbrace{D^{(k-1)}_{k,j}}_{\text{k~j까지}}}\right)$$

This (recursive) equation itself says how to compute $D^{(n)}$: use $D^{(k-1)}$ to compute $D^{(k)}$, for $k = 1, \ldots, n-1$. So we have a dynamic programming algorithm that computes $D^{(n)}$.

5/22(목)