30/03/2022

$$dist[u,v,l] := \begin{cases} 0 & \text{if } l=0 \text{ and } u=v \\ \infty & \text{if } l=0 \text{ and } v\neq u \\ \min \begin{cases} dist[u,v,l-1], \\ \min\limits_{x\to v} \{dist[u,x,l-1] + w(x\to v)\} \end{cases} \end{cases}$$

$w(u\to u) = 0$ if (u,v) edge doesn't exist then $w(u\to v) = \infty$.

$$dist(u,v,l) = \min_{x\in V} \{ dist[u,x,l-1] + w(x\to v) \}$$

$$\min \left\{ \begin{array}{ll} \text{if } x=v, & dist[u,v,l-1], \\ \text{if } x\to v, & dist[u,x,l-1] + w(x\to v), \\ \text{if } x\not\to v. & dist[u,x,l-1] + \infty \end{array} \right\}$$

$A_{n\times n}$ and $B_{n\times n}$ two matrices.

$$C_{ij} = \sum_{k=1}^{n} A_{ik}\cdot B_{kj}.$$

$\sum$:= min

Product := $+$

$$dist(u, v, l) = \min_{x \in V} \{ dist[u, x, l-1] + w(x \to v) \}$$

$$dist[l-1] = \begin{array}{c c} & v \\ u & \begin{bmatrix} & \\ & \bullet \\ & \end{bmatrix}_{n \times n} \end{array} \qquad W = \begin{bmatrix} 0 & & & & \\ & 0 & & & \\ & & 0 & & \\ & & & 0 & \\ & & & & 0 \end{bmatrix}_{n \times n}$$

$dist[u, v, l-1]$ = length of the shortest path from $u \to v$ using at most $(l-1)$ edges.

$$dist[l] = dist[l-1] \bullet W$$

$\bullet$ min-plus product.

You can multiply two matrices in min-plus product using at $O(n^3)$ operations / time.

$$APSP := W^{n-1}$$

iterated min-plus product (tropical product)

$dist[l=1] = W$, $dist[l=2] = W^2$, ... and so on.

→ to obtain W^{n-1}, you do repeated squaring $W, W^2, W^4, W^8, \dots, W^{n-1}$

Floyd - Warshall algorithm

$|V| = n, \quad |E| = m.$

→ Suppose your vertices are numbered from 1 to n.

① Shortest path $u \rightsquigarrow v$ using at most $(n-1)$ edges

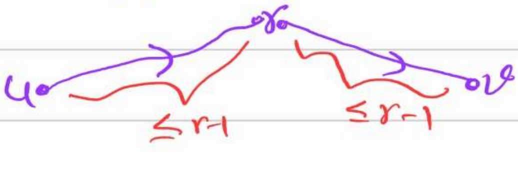
② Shortest path $u \rightsquigarrow v$ using vertices from $\{1, \dots, n\}$

* that passes through vertices from $\{1, \dots, n\}$.

$\pi(u, v, r)$:= Shortest path from $u \rightsquigarrow v$ using vertices with number $\leq r$.

$\pi(u, v, r) \xrightarrow{\text{the vertex } r \text{ is not used}} \pi(u, v, r-1)$

$\pi(u, v, r) \longrightarrow \pi(u, r, r-1) + \pi(r, v, r-1)$



$dist[u, v, r]$ = length of the shortest path from $u \rightsquigarrow v$ using vertices with number $\leq r$.

$$dist[u, v, r] = \begin{cases} w(u \to v) & \text{if } r = 0 \\ \min \left\{ dist[u, v, r-1],\ dist[u, r, r-1] + dist[r, v, r-1] \right\} & \end{cases}$$

Floyd-Warshall algo

For all vertices u
  For all vertices v
    $dist[u, v, 0] := w(u \to v)$

For $r = 1$ to n
  For all vertices u
    For all vertices v
      $dist[u, v, r] = \min \left\{ dist[u, v, r-1],\ dist[u, r, r-1] + dist[r, v, r-1] \right\}$
    End for
  End for
End for

Output $dist[u, v, n]$ $\forall$ u, v.

Runtime : $O(n^3)$

Jhonson's Algo :- $O(nm \log n)$

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