Shortest Paths From All Vertices To All Vertices

Using the Warshall-type algorithm to compute the ~~distance~~ cost matrix, we will complete with a P matrix to get the paths too, not only the distances.

The Floyd-Warshall algorithm , no negative costs

Initially $p_{ij} := i$ if $d_{ij} \neq \infty$ and $i \neq j$, and in other cases $p_{ij} := 0$.

$D := D_0$
for $k := 1$ **to** n **do**
    for $i := 1$ **to** n **do**
        for $j := 1$ **to** n **do**
            if $d_{ij} > d_{ik} + d_{kj}$ **then** , complexity $O(|V|^3)$
              $d_{ij} := d_{ik} + d_{kj}$
              $p_{ij} := p_{kj}$
            endif
        endfor
    endfor
endfor

$D_0, D_1, D_2, \dots D_{k-1} = D_k = D$

$D_0 := (d_{ij}^{(0)})_{i,j=\overline{1,n}}$

$$\text{where } d_{ij}^{(0)} = \begin{cases} \overset{\text{cost}}{W}(v_i, v_j) & \text{if } \{v_i, v_j\} \in E \\ 0 & i = j \\ \infty & \text{if } \{v_i, v_j\} \notin E, i \neq j \end{cases}$$

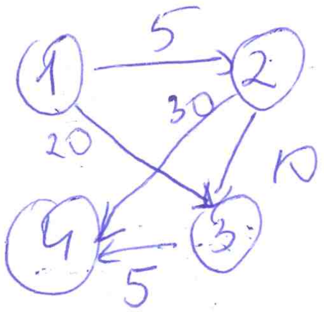
For $k > 0$: $d_{ij}^{(k)} := \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$ for $i, j = 1, 2, \dots, n$.

An x–y paths is determined by the following algorithms:

$k := n:$
$u_k := y$
while $u_k \neq x$ **do**
    $u_{k-1} := p_{xu_k}$
    $k := k - 1$
endwhile

The path is: $u_k, u_{k+1}, \dots, u_n$

Floyd-Warshall alg: from all to all vertices



D_i = cost matrices , P_i - previous matrices

$$D_0 = \begin{pmatrix} 0 & 5 & 20 & \infty \\ \infty & 0 & 10 & 30 \\ \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad P_0 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$k=1 \to$ using vertex: 1 as an intermediate vertex

$$D_1 = \begin{pmatrix} 0 & 5 & 20 & \infty \\ \infty & 0 & 10 & 30 \\ \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad P_1 = \begin{pmatrix} 0 & 1 & 1 & 0 \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$k=2 \to$ using vertex: 2 as an intermediate vertex

$$D_2 = \begin{pmatrix} 0 & 5 & \boxed{15} & \boxed{35} \\ \infty & 0 & 10 & 30 \\ \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix} \qquad P_2 = \begin{pmatrix} 0 & 1 & \boxed{2} & \boxed{2} \\ 0 & 0 & 2 & 2 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$k=3 \to$ using vertex: 3 as an intermediate vertex

$$D_3 = \begin{pmatrix} 0 & 5 & 15 & \boxed{20} \\ \infty & 0 & 10 & \boxed{15} \\ \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix}, \qquad P_3 = \begin{pmatrix} 0 & 1 & 2 & \boxed{3} \\ 0 & 0 & 2 & \boxed{3} \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

$k=4 \to$ using vertex: 4 as an intermediate vertex

$$D_4 = \begin{pmatrix} 0 & 5 & 15 & 20 \\ \infty & 0 & 10 & 15 \\ \infty & \infty & 0 & 5 \\ \infty & \infty & \infty & 0 \end{pmatrix}, \qquad P_4 = \begin{pmatrix} 0 & 1 & 2 & 3 \\ 0 & 0 & 2 & 3 \\ 0 & 0 & 0 & 3 \\ 0 & 0 & 0 & 0 \end{pmatrix} \quad \underline{|V|=4} \to \text{stop.}$$

$D_4(3,2) = \infty \Rightarrow$ there is no walk from 3 to 2.

The minimum cost walk from $s=1$ to $t=4$ has the cost: $\boxed{D_4(1,4) = 20}$

and it is obtained from P_4 backwards, using line $1=s$:

$\underline{t=4}$, $P_4(1,4) = \underline{3}$, $P_4(1,3) = \underline{2}$, $P_4(1,2) = \underline{1 = s}$

The minimum cost walk : $1 \xrightarrow{5} 2 \xrightarrow{10} 3 \xrightarrow{5} 4$ ::

Shortest Paths From All Vertices To All Vertices

Using the Warshall-type algorithm to compute the distance matrix, we will complete with a P matrix to get the paths too, not only the distances.

The Floyd-Warshall algorithm

Initially $p_{ij} := i$ if $d_{ij} \neq \infty$ and $i \neq j$, and in other cases $p_{ij} := 0$.

$D := D_0$
for $k := 1$ **to** n **do**
  for $i := 1$ **to** n **do**
    for $j := 1$ **to** n **do**
      if $d_{ij} > d_{ik} + d_{kj}$ **then**
        $d_{ij} := d_{ik} + d_{kj}$
        $p_{ij} := p_{kj}$
      endif
    endfor
  endfor
endfor

complexitate: $O(|V|^3)$

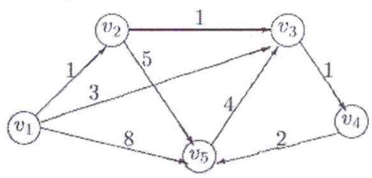
$D_0, D_1, D_2, \ldots D_{k-1} = D_k = D$

$D_0 := (d_{ij}^{(0)})_{i,j=\overline{1,n}}$

where $d_{ij}^{(0)} = \begin{cases} \mathcal{W}(v_i, v_j) & \text{if } \{v_i, v_j\} \in E \\ 0 & i = j \\ \infty & \text{if } \{v_i, v_j\} \notin E, i \neq j \end{cases}$

For $k > 0$: $d_{ij}^{(k)} := \min\left(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\right)$ for $i, j = 1, 2, \ldots, n$.

Example.



The adjacency matrix of the weighted graph given in the above figure:

$$D_0 = \begin{pmatrix} 0 & 1 & 3 & \infty & 8 \\ \infty & 0 & 1 & \infty & 5 \\ \infty & \infty & 0 & 1 & \infty \\ \infty & \infty & \infty & 0 & 2 \\ \infty & \infty & 4 & \infty & 0 \end{pmatrix}$$

The distance matrix D and the matrix P of previous vertices:

$$D = \begin{pmatrix} 0 & 1 & 2 & 3 & 5 \\ \infty & 0 & 1 & 2 & 4 \\ \infty & \infty & 0 & 1 & 3 \\ \infty & \infty & 6 & 0 & 2 \\ \infty & \infty & 4 & 5 & 0 \end{pmatrix} \qquad P = \begin{pmatrix} 0 & 1 & 2 & 3 & 4 \\ 0 & 0 & 2 & 3 & 4 \\ 0 & 0 & 0 & 3 & 4 \\ 0 & 0 & 5 & 0 & 4 \\ 0 & 0 & 5 & 3 & 0 \end{pmatrix}$$