... of shortest paths from a fixed vertex s to all other vertices in G.

10.5.3 Floyd-Warshall Algorithm

Notations in the algorithm:

1. Assume that the graph under consideration is represented by an $n \times n$ matrix $[w_{ij}]$ defined by

$$w_{ij} = \begin{cases} 0, & \text{if } v_i = v_j; \\ w(v_i v_j), & \text{if } v_i \neq v_j; \\ \infty, & \text{if } v_i v_j \notin E;. \end{cases}$$

2. D is the matrix defined by $D = [d_{ij}]$, where d_{ij} is the distance between the vertices v_i and v_j.
3. Let $d_{ij}^{(k)}$ be the length of the shortest path from v_i to v_j such that all intermediate vertices on the path (if any) are in set $\{v_1, v_2, \ldots, v_k\}$. Then, $d_{ij}^{(0)} = w_{ij}$, that is, no intermediate vertex between v_i and v_j. Also, let $D^{(k)} = [d_{ij}^{(k)}]$, where $1 \leq k \leq n$.

10.5.4 Floyd-Warshall Algorithm

S-1 : Initailly set $D \leftarrow D^{(0)}$. Set $i = 1, j = 1$ and set $k = 1$.

S-2 : Let $D^{(k)} = [d_{ij}^{(k)}]$ such that

(a) if v_k is not on a (the) shortest $v_i - v_j$ path, then it has length $d_{ij}^{(k-1)}$.

(b) if v_k is on a path, then it consists of a subpath from v_i to v_k and a subpath from v_k to v_j and must be as short as possible, namely they have lengths $d_{ik}^{(k-1)}$ and $d_{kj}^{(k-1)}$. Then, the shortest $v_i - v_j$ path has length $d_{ik}^{(k-1)} + d_{kj}^{(k-1)}$.

Combining the two cases we get, $d_{ij}^{(k)} = \min\{d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)}\}$.

S-3 : If $D = D^{(n)}$, then print D. Stop.