

Given a graph $G = (V, E)$, a cost function $c(i, j)$, a source node s and a destination node d , the algorithm computes the shortest (i.e. minimum-cost) path from s to d . Computation is performed by calling the recursive function $L(s, d, \mathbb{S})$. \mathbb{S} stands for “list of already visited nodes”, the initial value should be V .

$$\begin{aligned}
c(i, j) &= \begin{cases} +\infty & \text{if } i \text{ not connected to } j \\ \text{edge cost from } i \text{ to } j & \text{otherwise} \end{cases} \\
\operatorname{argmin}_2(I, f(i)) &= (\min_{i \in I} f(i), \operatorname{argmin}_{i \in I} f(i)) \\
I(\mathbb{S}, d) &= i \in \{ \delta^-(d) \cap \delta^+(\mathbb{S} \setminus \{d\}) \} \\
L(s, d, \mathbb{S}) &= \begin{cases} s = d & \implies (0, \phi) \\ \text{else} & \begin{cases} I(\mathbb{S}, d) = \phi & \implies (+\infty, \phi) \\ \text{else} & \operatorname{argmin}_2(I(\mathbb{S}, d), f(i) \doteq L(s, i, \mathbb{S} \setminus \{d\}) + c(i, d)) \end{cases} \end{cases}
\end{aligned}$$