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 $L(s, d, \mathbb{S})$.

$$\begin{aligned}
c(i, j) &= \begin{cases} +\infty & if\, i\, not\, connected\, to\, j \\ edge\, cost\, from\, i\, to\, j & otherwise \end{cases} \\
argmin_2(I, f(i)) &= (min_{i \in I}\, f(i),\, argmin_{i \in I}\, f(i)) \\
I(\mathbb{S}, d) &= i \in \{\, \delta^-(d) \cap \delta^+(\mathbb{S} \setminus \{d\})\, \} \\
L(s, d, \mathbb{S}) &= \begin{cases} s = d & \implies (0, \phi) \\ else & \begin{cases} I(\mathbb{S}, d) = \phi & \implies (+\infty, \phi) \\ else & argmin_2(\, I(\mathcal{S}, d),\, f(i) \doteq L(s, i, \mathbb{S} \setminus \{d\}) + c(i, d)\,)\,) \end{cases} \end{cases}
\end{aligned}$$