

Dijkstra's Algorithm

The algorithm below works on a weighted, directed graph G , represented in an adjacency list. This algorithm takes in the graph G , a starting vertex s , and a weight function w as parameters. It returns nothing, but instead updates the d (distance) and π (predecessor) values of every vertex.

Dijkstra's is a greedy algorithm whose job it is to find the shortest path from a source vertex s to all vertices reachable from s . It is one of a handful of single-source shortest-paths (SSSP) algorithms, and works only when the graph contains non-negative weight edges. It uses a helper function, RELAX, to update the d and π values on the vertices. It also uses a helper function INITIALIZE-SINGLE-SOURCE, to initialize the d and π values before the algorithm begins.

DIJKSTRA(G, s, w)

```

1  INITIALIZE-SINGLE-SOURCE( $G, s$ )
2   $S = \emptyset$ 
3   $H = \emptyset$ 
4  for each vertex  $u \in G.V$ 
5      INSERT( $H, u$ )
6  while  $H \neq \emptyset$ 
7       $u = \text{EXTRACT-MIN}(H)$ 
8       $S = S \cup \{u\}$ 
9      for each vertex  $v \in G.Adj[u]$ 
10         RELAX( $u, v, w$ )
11         if the call of RELAX decreased  $v.d$ 
12             DECREASE-KEY( $H, v, v.d$ )
```

INITIALIZE-SINGLE-SOURCE(G, s)

```

1  for each vertex  $v \in G.V$ 
2       $v.d = \infty$ 
3       $v.\pi = \text{NIL}$ 
4   $s.d = 0$ 
```

RELAX(u, v, w)

```

1  if  $v.d > u.d + w(u, v)$ 
2       $v.d = u.d + w(u, v)$ 
3       $v.\pi = u$ 
```

We typeset the procedures above with the following L^AT_EX:

```

\begin{codebox}
\Procname{$\proc{Dijkstra}(G, s, w)$}
\li $\proc{Initialize-Single-Source}(G, s)$
\li $$ \gets \emptyset$
```

```

\li $H \gets \emptyset$
\li \For each vertex $u$ \in $G.V$
\Do
\li $\text{\proc{Insert}}(H, u)$
\End
\li \While $H \ne \emptyset$
\Do
\li $u \gets \text{\proc{Extract-Min}}(H)$
\li $S \gets S \cup \{u\}$
\li \For each vertex $v$ \in $G.Adj[u]$
\Do
\li $\text{\proc{Relax}}(u, v, w)$
\li \If the call of $\text{\proc{Relax}}$ decreased $v.d$
\Then
\li $\text{\proc{Decrease-Key}}(H, v, v.d)$
\end{codebox}

```

```

\begin{codebox}
\Procname{$\text{\proc{Initialize-Single-Source}}(G, s)$}
\li \For each vertex $v$ \in $G.V$
\Do
\li $v.d \gets \infty$
\li $v.\pi \gets \text{\const{Nil}}$
\End
\li $s.d \gets 0$
\end{codebox}

```

```

\begin{codebox}
\Procname{$\text{\proc{Relax}}(u, v, w)$}
\li \If $v.d > u.d + w(u, v)$
\Then
\li $v.d \gets u.d + w(u, v)$
\li $v.\pi \gets u$
\end{codebox}

```