

Floyd-Warshall Algorithm

The Floyd-Warshall algorithm is a bottom-up dynamic programming solution to finding the shortest path between every pair of vertices in a directed, weighted graph. It assumes that negative-weight edges may exist, but there are no negative-weight cycles. It accepts as input a an $n \times n$ matrix W . An entry $W[i, j]$ will be $w(i, j)$ if there is an edge from i to j , 0 if $i == j$, and ∞ if $i \neq j$ and there is no edge from i to j

FLOYD-WARSHALL(G, w)

```

1  let  $D^{(0)}$  be a new  $V \times V$  matrix
2  for  $i = 1$  to  $V$ 
3      for  $j = 1$  to  $V$ 
4          if  $i == j$ 
5               $d_{ij}^{(0)} = 0$ 
6          elseif  $(i, j) \in G.E$ 
7               $d_{ij}^{(0)} = w(i, j)$ 
8          else
9               $d_{ij}^{(0)} = \infty$ 
10 for  $k = 1$  to  $V$ 
11     let  $D^{(k)}$  be a new  $V \times V$  matrix
12     for  $i = 1$  to  $V$ 
13         for  $j = 1$  to  $V$ 
14              $d_{ij}^{(k)} = \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
15 return  $D^{(V)}$ 
```

We typeset the Floyd-Warshall procedure above with the following L^AT_EX:

```

\begin{codebox}
\Procname{$\proc{Floyd-Warshall}(G, w)$}
\li let  $D^{(0)}$  be a new  $V \times V$  matrix
\li \For  $i$  \gets 1 \To  $V$ 
\Do
\li \For  $j$  \gets 1 \To  $V$ 
\Do
\li \If  $i == j$ 
\Then
\li  $d_{ij}^{(0)} = 0$ 
\li \ElseIf  $(i, j) \in G.E$ 
\Then
\li  $d_{ij}^{(0)} = w(i, j)$ 
\li \Else
\li  $d_{ij}^{(0)} = \infty$ 
\End
```

```

\End
\End
\li \For $k \gets 1 \To V$
\Do
\li let  $D^{(k)}$  be a new  $V \times V$  matrix
\li \For $i \gets 1 \To V$
\Do
\li \For $j \gets 1 \To V$
\Do
\li  $d_{ij}^{(k)} \gets \min(d_{ij}^{(k-1)}, d_{ik}^{(k-1)} + d_{kj}^{(k-1)})$ 
\End
\End
\End
\li \Return  $D^{(V)}$ 
\end{codebox}

```