

## Topological Sort

The algorithm below creates a topological sort of a directed, acyclic graph. A topological sort is a linear ordering of all the vertices such that if  $G$  contains an edge  $(u, v)$  then  $u$  appears before  $v$  in that ordering.

This algorithm relies on Depth-First Search, which is a cousin of Breadth-First Search. Instead of visiting all the neighbors of a given node (breadth), DFS explores as far down a path as it possibly can (depth), and backtracks after it has exhausted a path.

TOPOLOGICAL-SORT( $G$ )

- 1 call DFS( $G$ ) to compute finishing times  $v.f$  for each vertex  $v$
- 2 **return** the vertices in descending order of finish time

The run-time of Topological sort is  $\Theta(V \lg V)$  because it is bound by the sort of all the finish times. The run-time of DFS on its own is  $\Theta(V + E)$ .

DFS( $G$ )

- 1 **for** each vertex  $u \in G.V$
- 2      $u.color = \text{WHITE}$
- 3      $u.\pi = \text{NIL}$
- 4  $time = 0$
- 5 **for** each vertex  $u \in G.V$
- 6     **if**  $u.color == \text{WHITE}$
- 7         DFS-VISIT( $G, u$ )

DFS-VISIT( $G, u$ )

- 1  $time = time + 1$
- 2  $u.s = time$
- 3  $u.color = \text{GRAY}$
- 4 **for** each  $v \in G.Adj[u]$
- 5     **if**  $v.color == \text{WHITE}$
- 6          $v.\pi = u$
- 7         DFS-VISIT( $G, v$ )
- 8  $u.color = \text{BLACK}$
- 9  $time = time + 1$
- 10  $u.f = time$

We typeset the procedures above with the following  $\text{\LaTeX}$ :

```
\begin{codebox}
\Procname{\$\proc{Topological-Sort}(G)\$}
```

```

\li call  $\text{\proc{DFS}}(G)$  to compute finishing times  $v.f$  for each vertex  $v$ 
\li \Return the vertices in descending order of finish time
\end{codebox}

```

```

\begin{codebox}
\Procname{ $\text{\proc{DFS}}(G)$ }
\li \For each vertex  $u$  \in  $G.V$ 
\Do
\li  $u.\text{color}$  \gets \const{White}
\li  $u.\pi$  \gets \const{Nil}
\End
\li  $\text{id}\{time\}$  \gets 0
\li \For each vertex  $u$  \in  $G.V$ 
\Do
\li \If  $u.\text{color} == \text{const}\{White\}$ 
\Then
\li  $\text{\proc{DFS-Visit}}(G, u)$ 
\end{codebox}

```

```

\begin{codebox}
\Procname{ $\text{\proc{DFS-Visit}}(G, u)$ }
\li  $\text{id}\{time\}$  \gets  $\text{id}\{time\} + 1$ 
\li  $u.s$  \gets  $\text{id}\{time\}$ 
\li  $u.\text{color}$  \gets \const{Gray}
\li \For each  $v$  \in  $G.\text{Adj}[u]$ 
\Do
\li \If  $v.\text{color} == \text{const}\{White\}$ 
\Then
\li  $v.\pi$  \gets  $u$ 
\li  $\text{\proc{DFS-Visit}}(G, v)$ 
\End
\End
\li  $u.\text{color}$  \gets \const{Black}
\li  $\text{id}\{time\}$  \gets  $\text{id}\{time\} + 1$ 
\li  $u.f$  \gets  $\text{id}\{time\}$ 
\end{codebox}

```