Topological Sort

- Given a DAG, directed acyclic graph
- Find an ordering of the vertices such that if \((v, w) \in E\) then \(v\) is before \(w\) in the ordering.

Algorithm

- DFS(G)
- Output the nodes in order of decreasing finishing times

Running time: \(O(E)\)
Proof of Correctness

**Theorem**  Topological Sort Algorithm is correct, i.e. if \((x, y) \in E\) then \(f(x) > f(y)\).

**Proof**

Case 1: \((d(x) < d(y))\)

- At time \(d(x)\), \(y\) must be white.
- Using the parenthesis theorem,

\[d(x) < d(y) < f(y) < f(x).\]

Case 2: \((d(x) > d(y))\).

- Because \(G\) is a DAG and we have an edge \((x, y)\), there is no path from \(y\) to \(x\).
- This means that \(f(y) < d(x)\).
- For any vertex \(d(x) < f(x)\).
- Putting these two inequalities together we get \(f(y) < f(x)\).