DFS numbering

pre-order number
post-order number

DFS numbering

many of them
1) where we start
2) how we visit the outgoing edges

Containment of intervals indicate ancestor-descendant relationships

Detecting cycle

DA6 := Directed Acyclic Graph

given a directed graph, is it a DA6 or not?
\[ \rightarrow \text{does it have a cycle or not?} \]
Given a directed graph, does it have a cycle or not?

**Observation:** If there is a cycle, for a node $u$ and $v$ such that

\[ [s_u, e_u] \supseteq [s_v, e_v] \]

such that

\[ \text{Interval of } u \subseteq \text{Interval of } v \]

\[ [s_v, e_v] \subseteq [s_u, e_u] \]

If at $v$, we find a link to some vertex $u$ such that $\text{Interval of } v \subseteq \text{Interval of } u$ (that has already been visited)

then $\equiv$ a cycle.
Alg to Detect Cycles:

1) Do DFS intervals \( O(N+M) \)

2) \( \forall u \in V \) for all \( x \in \text{neighbor}(u) \):
   - If interval \([x]\) \( \supseteq \) interval \([u]\):
     - Cycle found.

\( O(N+M) \) linear time

\( O(N+M) \) worse case

Topological sort

Dependency graph
all of the "pre-requsites" of a node must be before that node in the ordering.

Topo Sort:

1) Do a DFS numbering

2) output vertices in reverse order of the post order link

This is a topological sort

\[ O(V + E) \]

\[ O(V \log V) \]

\[ O(V + E + V \log V) \]

\[ O(V + E) \]
Proof of Correctness:

\[
\text{Interval } [u, v] \text{ contains all reachable node's intervals}
\]

Post \((u)\) must be larger than all other nodes that are reachable from \(u\).
Strongly connected components (SCC)

Directed graph (with cycle)

SCC: maximal set of vertices that are mutually reachable from each other

\[ \{x, y\} \in SCC \implies x \text{ can reach } y \quad \text{mutually reachable} \quad y \text{ can reach } x \]

\[ \text{we cannot add another vertex to the SCC} \]
Directed graph has an inherent DAG structure over SCC.

Any directed graph is a DAG over its strongly connected components.

**SCC Algorithm:**

Given a directed graph $G = (V, E)$, output all SCCs.

1. Identify successively all "sink" nodes.
2. From each sink, do a DFS to extract the SCC, remove it, and mark all vertices as visited.

$O(|V| + |E|)$

A source vertex in a directed graph has the highest
1) Do DFS numbering on reverse graph $G^R$

2) Just traverse in decreasing order of $post(v)$ in $G$ and find all reachable vertex from $v$ (Every such set is a SCC).