**Lemma.** Suppose when DFS visits $u$, there is a path from $u$ to $v$, and $v$ is not visited. Then $u$ is on stack if $u$ is visited.

$u = 3 \quad v = 4$

- at the time 3 is visited
- path $(3, 0, 4)$, 4 is not visited.
- However: when 4 is visited, 3 is not on stack.

**Proof:** by induction.

*Induction Hypothesis.* Suppose when DFS visits $u$, there is a path from $u$ to $v$ of length at most $l$, and $u$ is the only visited vertex on the path, then $u$ is on stack when $u$ is visited.

*Base case:* $l = 1$ ($u, v$) is an edge, $v$ is not visited when $u$ is visited.

*Induction Step:* Suppose $IH$ is true for $l = k$, consider a path of length $k+1$.

Using $IH$ when $w$ is visited, $u$ is on stack.

- $u \rightarrow v$ has already been visited.

**Case 1:** if $v$ is not visited, immediately call $DFS\_visit(w)$ so $v$ is still visited while $u$ is on stack.

**Case 2:** if $v$ is not visited, immediately call $DFS\_visit(w)$ so $v$ is still visited while $u$ is on stack.
when \((w,v)\) is considered

case 0 \(v\) has already been visited.

\(\text{DFS-visit}(v)\) happens between \(\text{DFS-visit}(w)\) and when \((w,v)\) is considered. \(u\) is on stack for the entire time interval.

\[
\begin{array}{c}
    u \\
    w \\
    \rightarrow \\
    \rightarrow \\
    \rightarrow \\
    \rightarrow \\
    v
\end{array}
\]

\text{case 2} \(v\) is not visited.

in this case, call \(\text{DFS-visit}(v)\) immediately, so \(u\) is still on stack when \(v\) is visited.

---

\text{Post-order:}

\[h \prec d \prec a \prec b \prec g \prec f\]

\text{Lemma: a reverse post-order is a valid topological sort.}

\text{Lemma: for every edge } (u,v), \text{ } u \text{ must be later than } v \text{ in post-order.}

\text{Proof: use proof by contradiction}

assume towards contradiction that there is an edge \((u,v)\) where \(u\) is before \(v\) in post order.

if \(u\) is visited before \(v\)

by base case of previous lemma

\(u\) is on stack when \(v\) is visited

\(u\) is after \(v\) in post order.

if \(u\) is visited after \(v\)

\(\text{visit } v \quad \text{visit } u \quad \text{DFS-visit}(u)\) returns \(\text{DFS-visit}(v)\) returns

is the only possible sequence of events.

This means when \(u\) is visited, \(v\) is on stack.

by DFS algorithm, there is a path from \(u\) to \(u\), but \((u,v)\) is also an edge

so the path from \(v\) to \(u\) + \((u,v)\) form a cycle.

this contradicts with the assumption that the graph is acyclic.