Topological Sort: Input = DAG  Output = ordering of vertx
so edges all go →

Algorithm: Run DFS(G)
order vertices by ↓ f

Want to show ordering vertx by ↓ f gives a valid topological sort
For all edges (u,v), f[u] > f[v]

Correctness: Suppose (u,v) ∈ G.E (and f[u] > f[v])
At time, either a) color[v] = WHITE
  Then there is a white path u → v
  So by WPT v is a descendant of u
  d[u] < d[v] < f[v] < f[u]

b) color[v] = BLACK
  Then d[u] < f[v] < d[w] < f[w]

c) color[v] = GRAY
  v in another of u
  Then (u,v) is a back edge
  Then is a cycle

Application: Longest path (and simple path with most edges) adj list
  1) Topo sort
  2) For each u ∈ G.V in reverse order of topo sort
     l[u] = max \( \max_{(u,v) \in G.E} (l[v]) \), 0
     \( O(V+E) \)
  3) return \( \max_{v \in G.V} l[v] \)
     \( O(V) \)
     \( O(V+E) \)
\[ \ell(v) = \text{# edges in longest path that start at } v \]

\[ 5 \quad 4 \quad 2 \quad 3 \quad 2 \quad 0 \quad 1 \quad 0 \]
Strongly Connected Components

SCC: maximal subset of verts $V'$ of directed graph s.t. $V, v_1, v_2 \in V'$

Algorithm:
1) Run DFS $(G)$ on $G$
2) Run DFS $(\bar{G})$, considering verts in order of $\bar{f}$
3) DFS-trees from 2) are SCCs (in top sort order)

$O(V+E)$

Adj List
Correctness

Lemma: If $C, C'$ are distinct SCCs with $C \neq C'$, $u \in C$, $v \in C'$ then $f(C) \neq f(C')$

Corr: If $C, C'$ distinct SCCs, $u \in C, v \in C', (u,v) \in E^T$ then $f(C) \neq f(C')$

Then: SCC algorithm is correct

Invariant: $1st$ $k$ trees are SCCs