Depth-First Search

DFS-VISIT(u)
mark u as processing
and add u to end of visited list
for each neighbors v of the current vertex u
if v still unseen then DFS-VISIT(v)
else if v is processing then found a cycle
mark u as finished
and add u to front of finished list

Cycle detection
Topological Sort

In a given directed acyclic graph, order vertices such that all edges go →

0 → 2 → 1 ← 3 4 6 ← 5

0 1 3 4 5 6

3 2 → 4 1 ← 5 0 → 6 → 7
Given a directed graph, find the longest simple path in the graph.

**NP-complete**

but easy on directed acyclic graph

1. top sort
2. in reverse order
   - of top sort
   - compute $l(u)$
   - $O(nm)$

The length of the longest simple path starting at $u$ is defined as:

$$l(u) = \begin{cases} 0 & \text{if } u \text{ has no outgoing edges} \\ \max \{ l(v) + 1 \} & \text{if there is an edge } (u, v) \end{cases}$$

For the given graph:

- $l(6) = \max \{ l(7) + 1 \}$
- $l(5) = \max \{ l(6) + 1 \}$
- $l(5) = \max \{ l(4) + 1 \}$
- $l(4) = \max \{ l(3) + 1 \}$
- $l(3) = \max \{ l(2) + 1 \}$
- $l(2) = \max \{ l(1) + 1 \}$
- $l(1) = \max \{ l(D) + 1 \}$
- $l(D) = 0$
Strongly Connected Components

In a directed graph, a maximal subset of vertices s.t. for all \( u, v \) in subset, there is a path \( u \rightarrow v \) and a path \( v \rightarrow u \).

\[
\begin{align*}
&\text{DFS: } 0 \ 3 \ 7 \ 6 \ 4 \ 1 \ 2 \ 5 \ 8 \\
&\text{run DFS on } G^r, \text{ pick starting points in reverse order of finish from } 0
\end{align*}
\]

Component graph directed acyclic graph

Topo sort of component graph