Task Scheduling

Precedence Graph for Changing a Flat Tire

- Drive off
  - Put lug tool away
    - Tighten lugs
      - Lower car
  - Put jack away
    - Put jack in trunk
      - Put on spare
        - Remove spare from trunk
        - Remove old tire
          - Jack up car
  - Put old tire in trunk
    - Remove lugs
      - Remove old tire
      - Place jack under car
      - Take out jack
    - Get out lug tool
      - Get out owner's manual
Topological Sort

Find an ordering of vertices in a directed acyclic graph so that if edge \( u \rightarrow v \) then \( u \) precedes \( v \) in ordering.

Should report a cycle if the graph has no valid order.
Observation: Valid topological order is defined by the reverse order of finishing times.
DFS visit call (W)

If edge \( u \rightarrow v \) we are looking at \( v \) is gray, there is a cycle.

Stack top \( u \)

When discovered, put on stack

When finished, taken off stack

\( v \), parent

\( v \), cycle
If there is a directed cycle, then at some point an edge will be encountered that goes gray vertex $u \rightarrow v$.

Cycle which first has discovered vertex

Stack

$u$

$V$ gray
If graph is acyclic, reverse order of finishing times is a valid topological order.

Suppose I've drawn in a valid order.