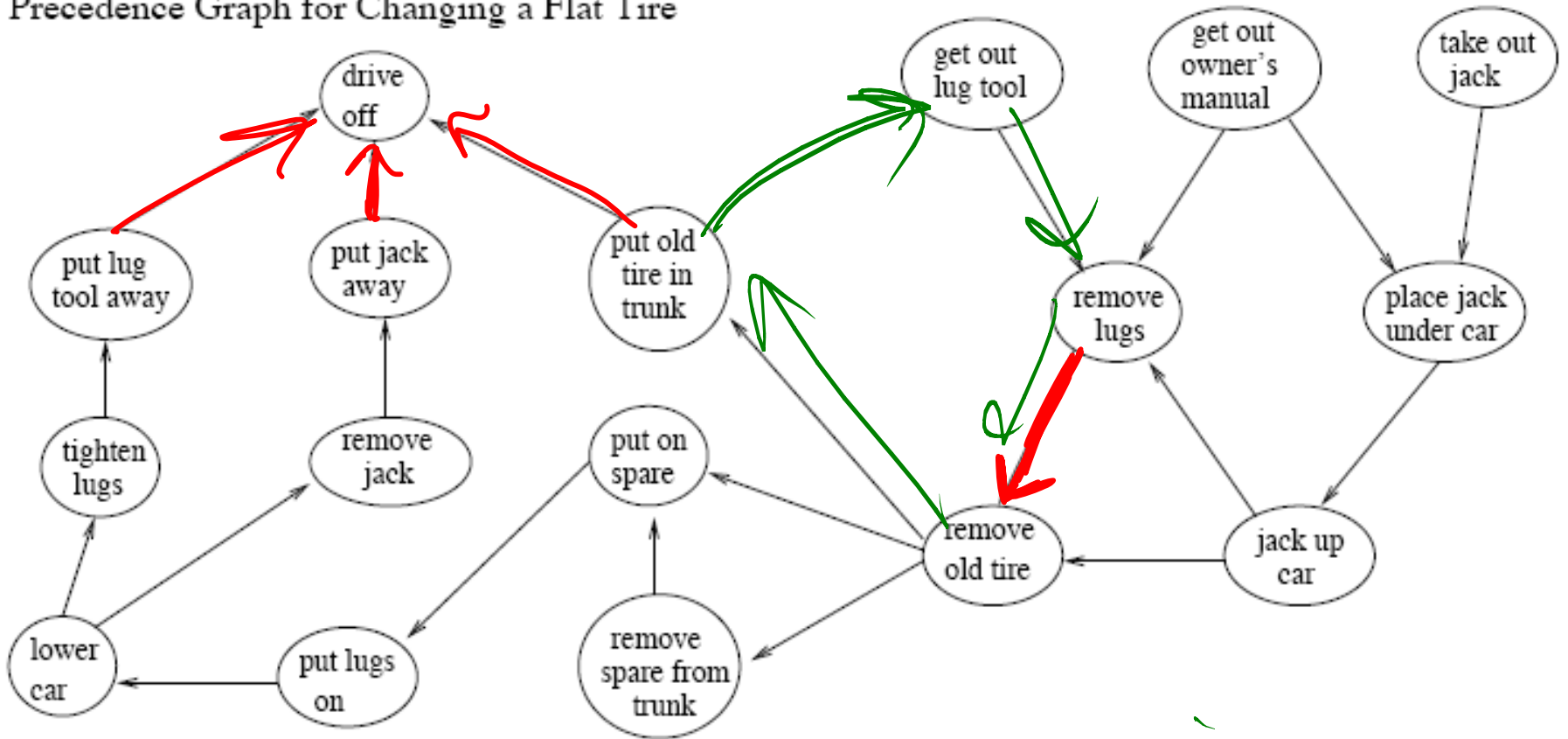


# Task Scheduling

$a \rightarrow b$   
a must precede b

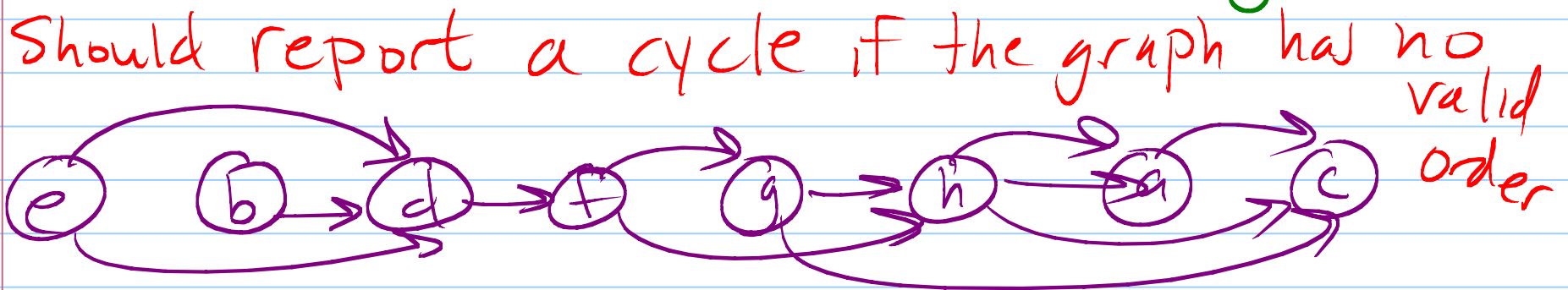
Precedence Graph for Changing a Flat Tire

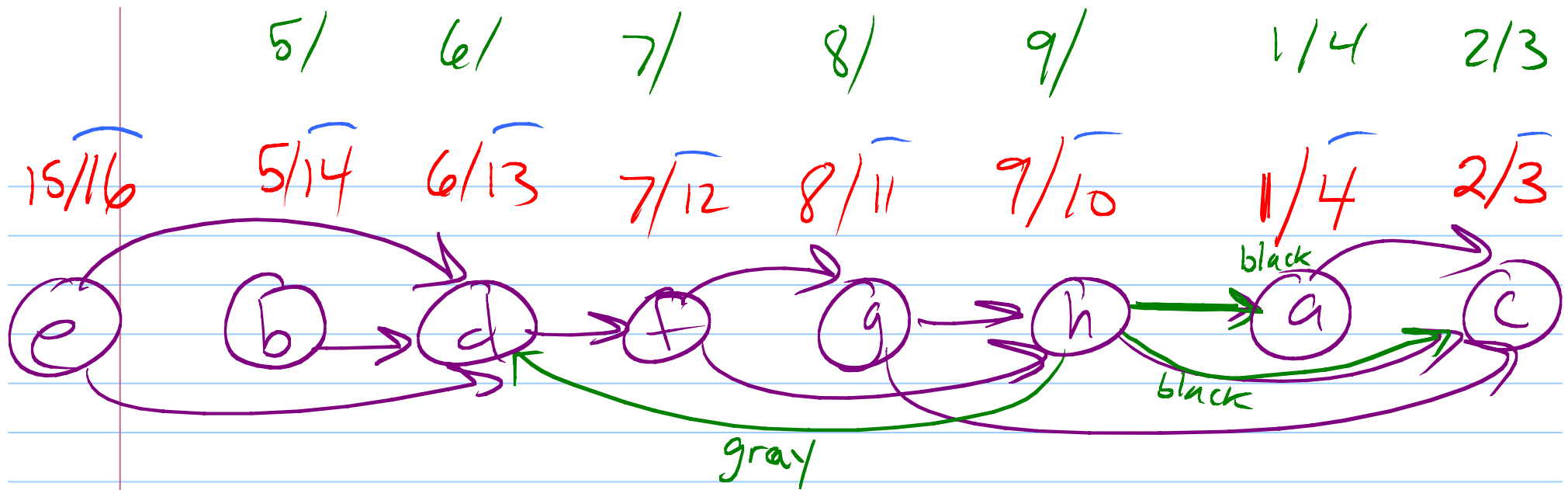


# Topological Sort

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

no directed cycles





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

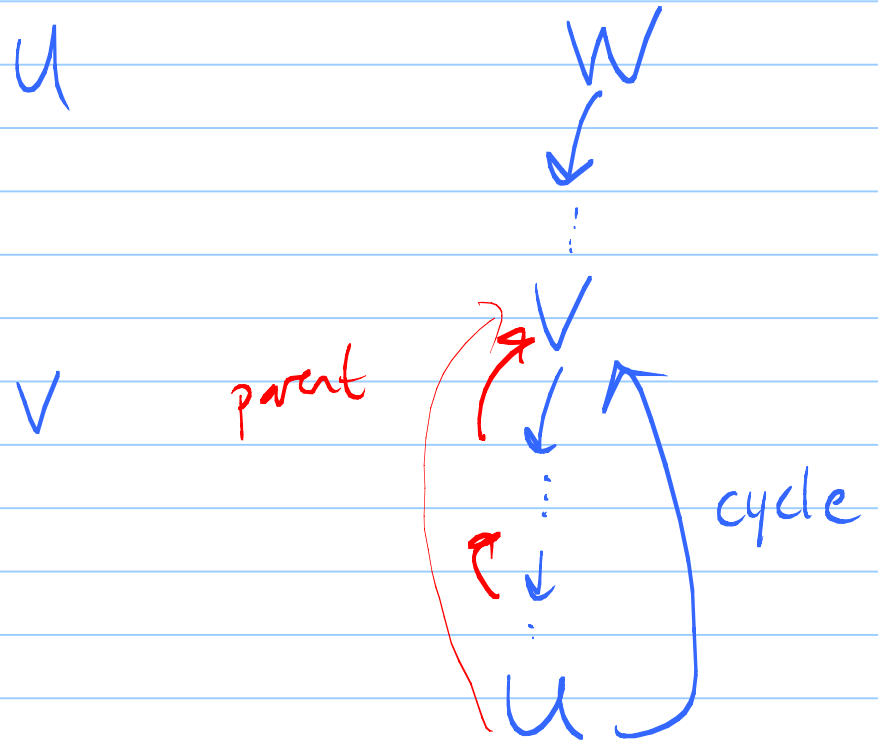
on  
stack

when  
discovered  
put on stack

when finished

taken off  
stack

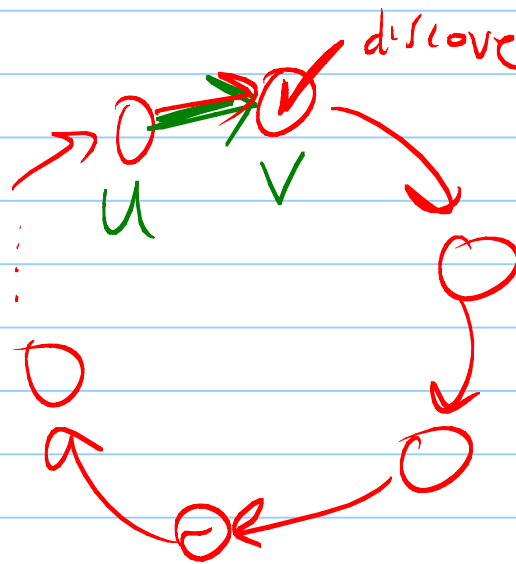
bottom



If there is a directed cycle,  
then at some point an edge  
will be encountered that goes  
gray vertex

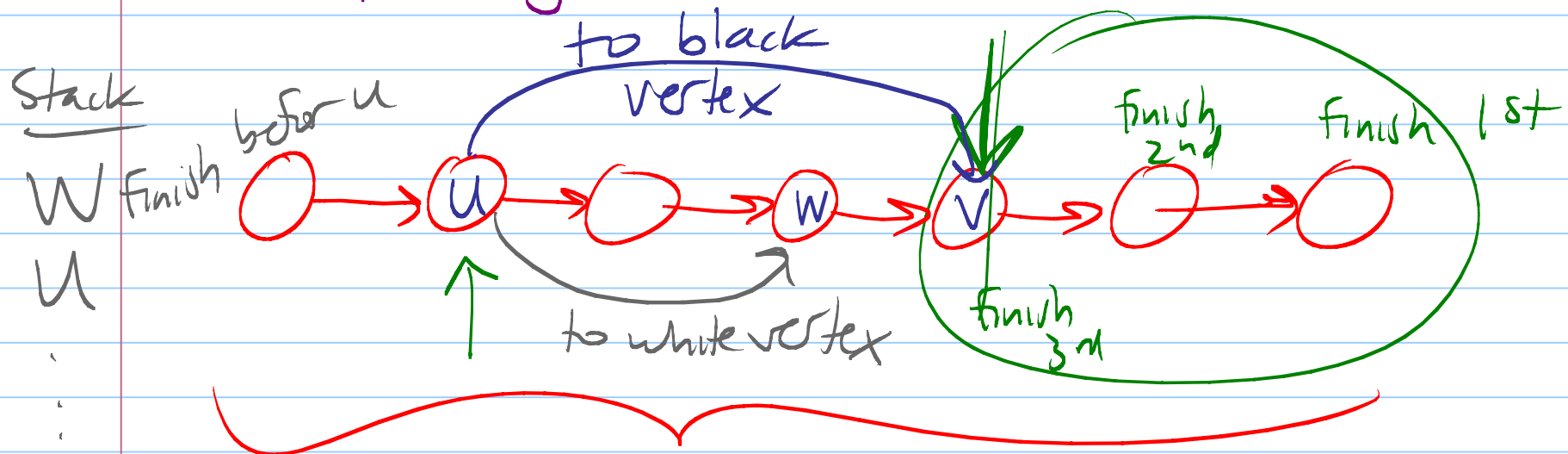
$u \rightarrow v$   
grey

cycle which  
first has  
discovered  
vertex



stack  
u  
v grey

IF graph is acyclic, reverse order of finishing times is a valid topological order



suppose I've drawn in a valid order