Depth-First Search

DFS-VISIT(u) ← called once per vertex reachable from start

mark u as processing
and add u to end of visited list

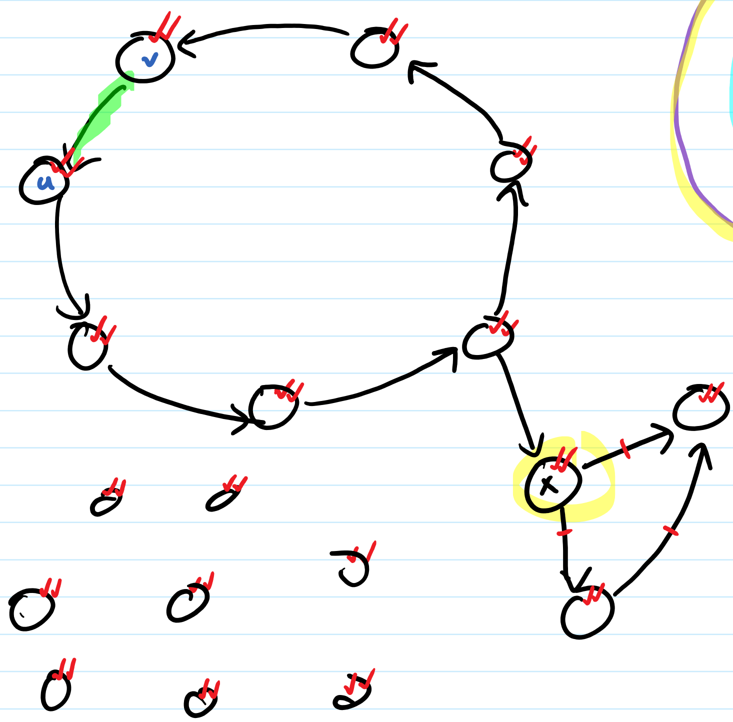
for each neighbors v of the current vertex u
    if v still unseen then DFS-VISIT(v)
    else if v is processing then found a cycle

mark u as finished
and add u to front of finished list

BFS $O(n+m)$ when using adj. list
#vertices #edges
($O(n^2)$ using adj. matrix)

→ for each vertex v
  for (out)neighbor of v
    do something

Cycle detection



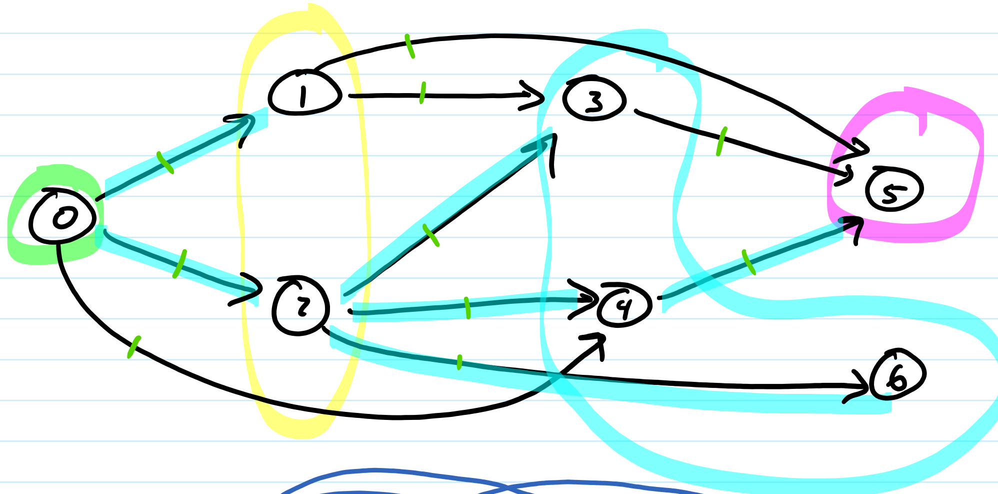
1st vertex on cycle ever passed to DFS-VISIT

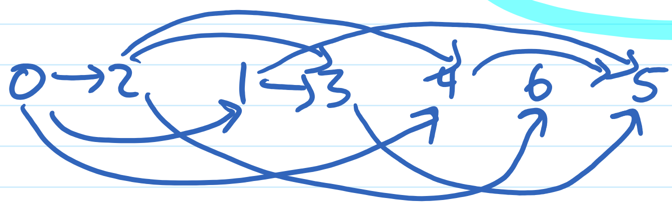
Processing DFS-VISIT(u)
⋮
DFS-VISIT(v)

DFS-VISIT(v_1) processing
DFS-VISIT(v_2) processing
DFS-VISIT(v_3) processing
⋮
DFS-VISIT(v_n) current

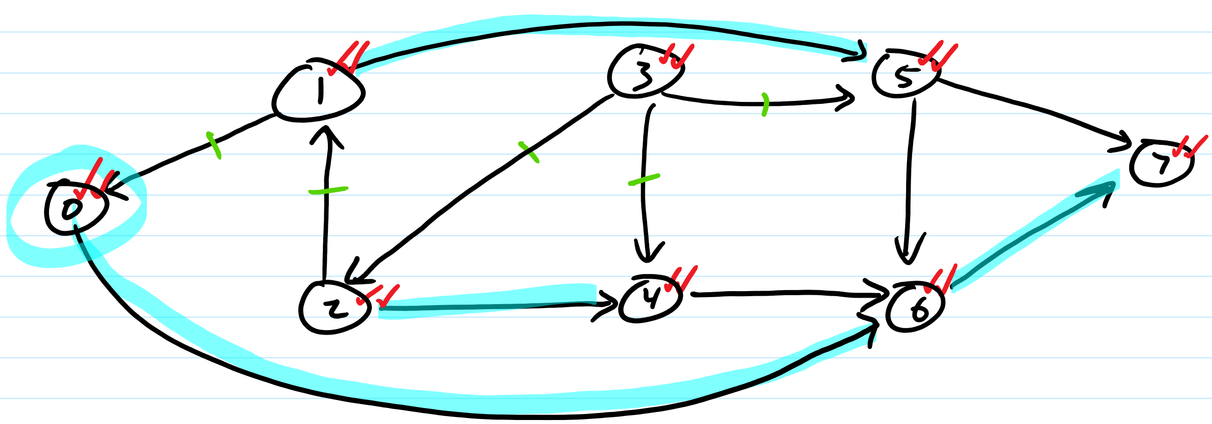
Topological Sort

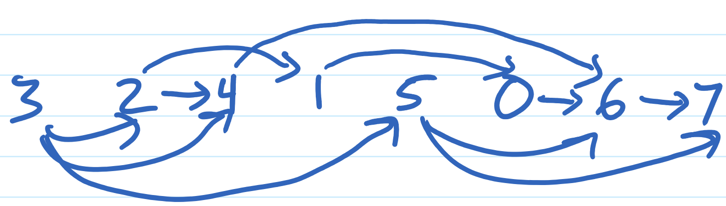
↳ given directed acyclic graph, order vertices s.t. all edges go →





0 1 3 4 5 6

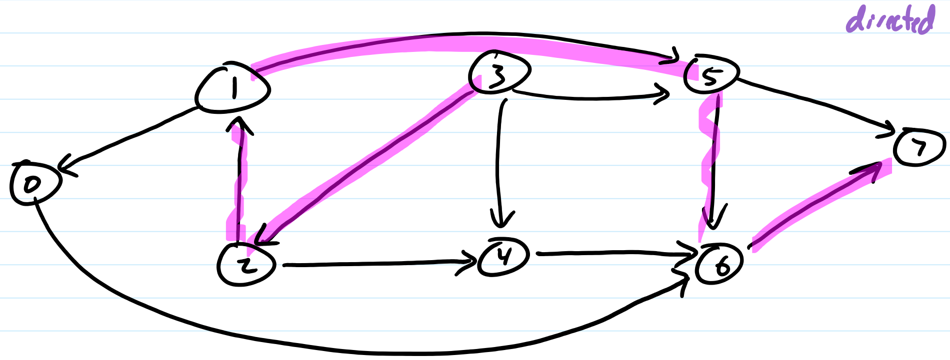




Longest Path

Given directed graph, find longest simple path in the graph
or Is there a simple path of length $\geq k$? ← NP-complete

but easy on directed acyclic graph



1) topo sort — $O(n+m)$
2) in reverse order of topo sort compute $\ell(u)$ — $O(n+m)$

$O(n+m)$

$\ell(u)$ = length of longest simple path starting at u

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

	3	2 →	4	1	5	0 →	6 →	7
$\ell(u)$	5	4	2	3	2	2	1	0

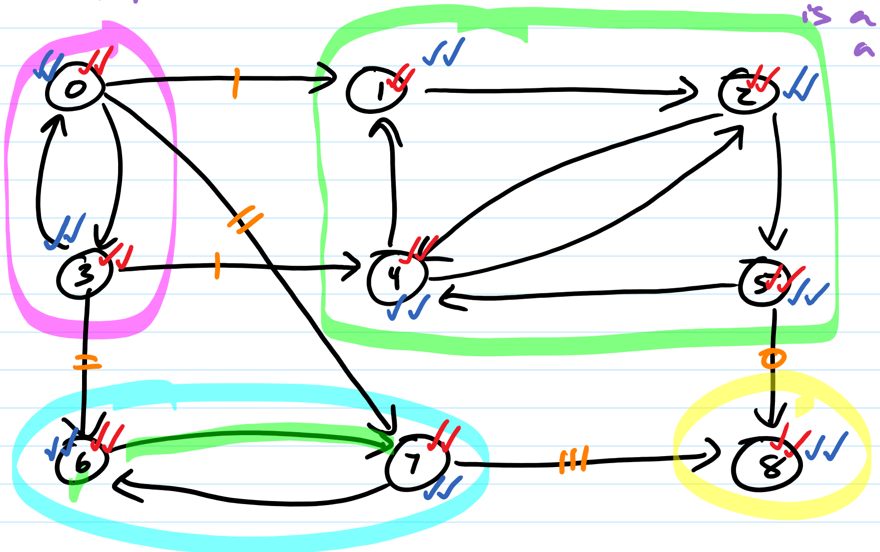
$\ell(6) = \max(\underbrace{\ell(7)+1}_{1})$

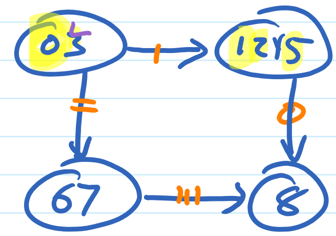
$\ell(0) = \max(\ell(6)+1)$

$\ell(5) = \max(\ell(\underset{2}{6})+1, \ell(\underset{1}{7})+1)$

Strongly Connected Components

↳ in directed graph, a maximal subset of verts. s.t. for all u,v in subset, there is a path u ⇝ v and a path v ⇝ u





component graph
directed acyclic graph

✓✓ 1) DFS   0 3 7 6 4 1 2 5 8

✓✓ 2) run DFS on G^T, pick starting points in reverse order of finish from 0
↳ G with edges reversed

8 | 4 5 2 1 | 7 6 | 0 3

topo sort of component graph