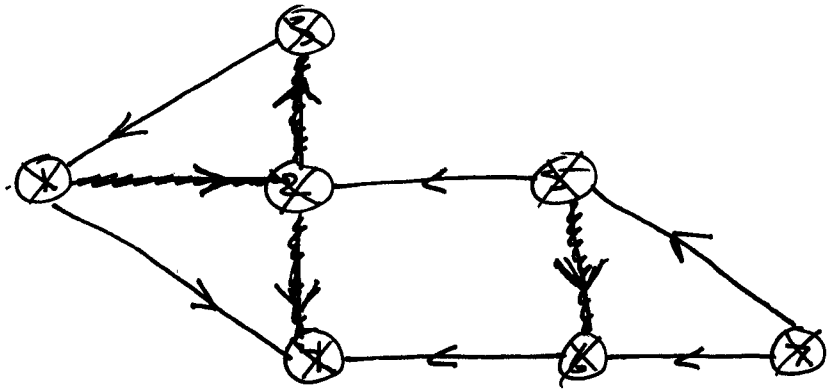
CSE 101 1-26-21

Quizz : today

Pa3 : ext. 1 day.

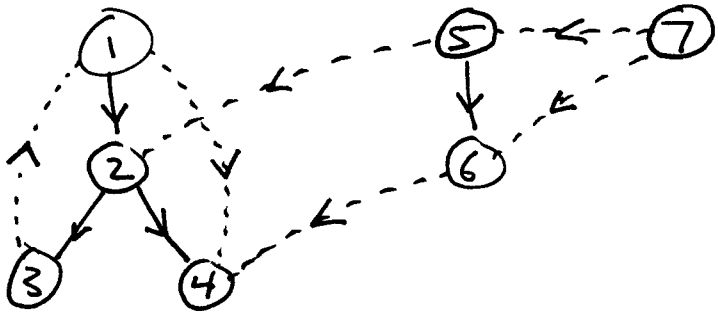
Ex



Edge classification

- tree (1,2), (2,3) (2,4), (5,6)
- back (3,1)
- forward : (1,4)
- cross : (5,2) (6,4), (7,5), (7,6)

DFS forest



Exercise

Run DFS on all prev. examples and classify edges. alter lables and Run DFS again, classifying edges.

Exercise

Modify DFS so that it prints each edge along with its classification.

Hint: pp. 609-610 (22.3-10) in CLRS (p-06.)

Thm

If DFS is run on an undirected graph, then there are no cross edges.

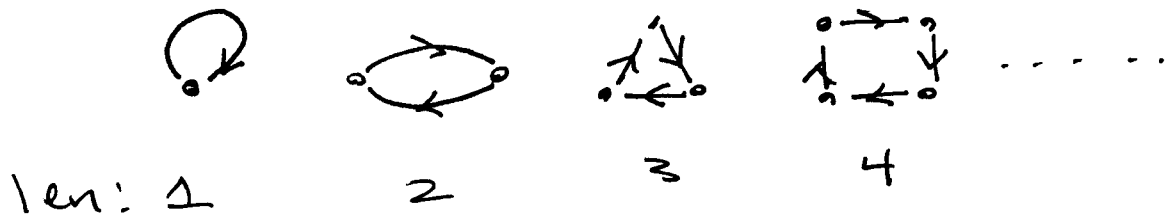
Topological Sort

Let $G = (V, E)$ be a digraph.

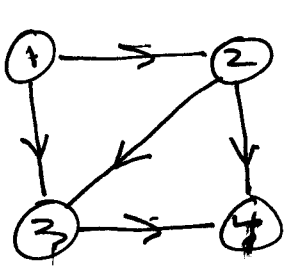
Defn

we say G is <u>acyclic</u> iff G contains no (directed) cycles.

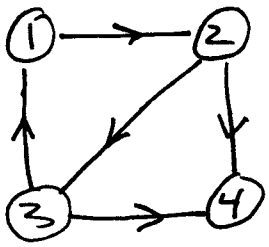
directed cycles:



Ex.



acyclic



not acyclic.

Lemma

a digraph G is acyclic iff DFS(G) yields no back edges.

equivalently

G contains a back edge iff G contains a directed cycle.

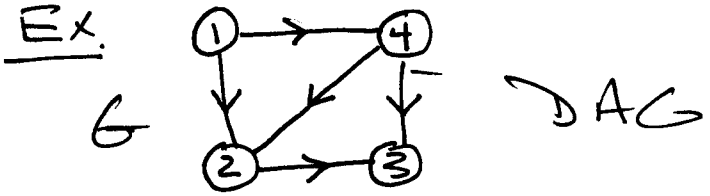
Defn

Let $G = (V, E)$ be a digraph. G is called a directed acyclic graph (DAG) iff it is acyclic.

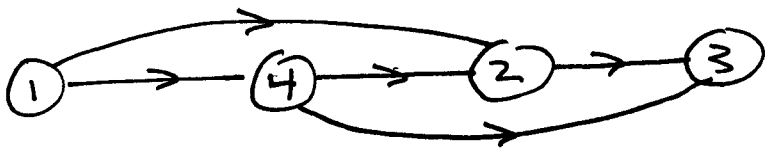
Defn

A topological sort of a DAG G is a linear ordering of $V(G)$ such that if $(x, y) \in E(G)$ then x is before y in the linear ordering.

Ex.



A Topological sort of G :

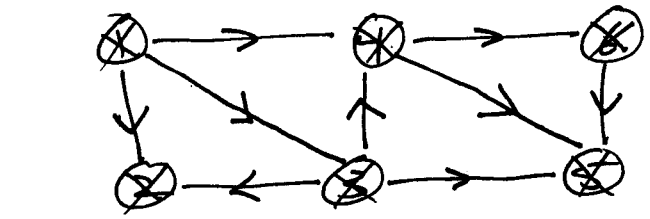


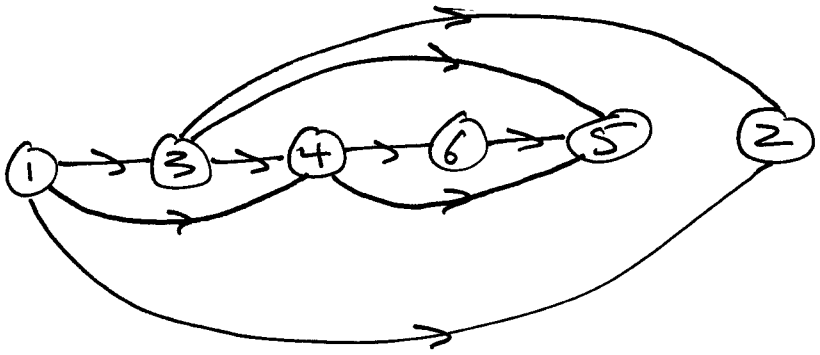
(p. 613 in CLRS)

To perform a topological sort on a DAG G:

- Run DFS(G)
- as vertices finish, push them onto a stack.
- when done the stack is a topological sort. (top to bottom.)

Ex





stack

1
3
4
6
5
2

Exercise:

The last example has 3 other topological sorts. find them.

Exercise

Permute labels and find the other Top. sorts using algorithm

strongly Connected Components (SCCs)

Let $G = (V, E)$ be a dig-aph. a subset $U \subseteq V$ is called strongly connected iff every vertex $x \in U$ is reachable from every vertex $y \in U$, and conversely.

Further, $U \subseteq V$ is called a strongly connected component (SCC) iff

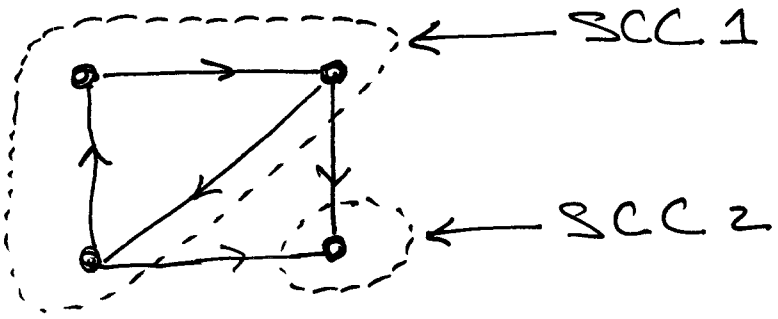
(1) U is strongly conn.

and

(2) U is maximal w.r.t (1),

i.e. whenever $U \subsetneq W \subseteq V$, then W is not strongly connected.

Ex.



To find the SCCs in a digraph

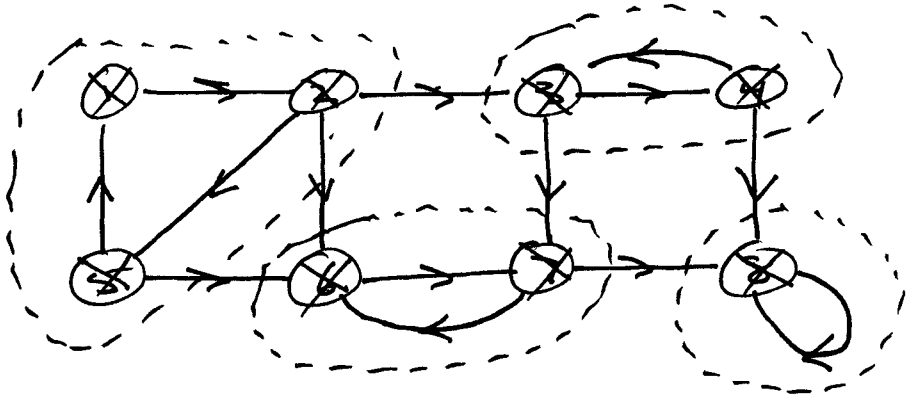
- Run DFS(G), as vertices finish push them onto a stack.
- Compute the transpose G^T of G (reverse all directions). i.e.

  $(x, y) \in E(G)$ iff $(y, x) \in E(G^T)$

- Run DFS(G^T), but process vertices in main loop by popping vertices off the stack. (equivalently: main loop is run by decreasing finish times from first Run of DFS.)
- When done, the vertices in each tree (from 2nd call) are the SCCs of G.

Ex

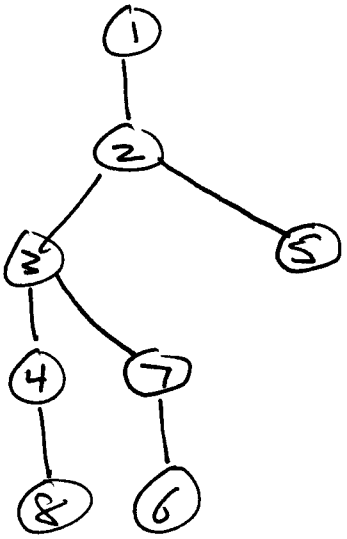
G



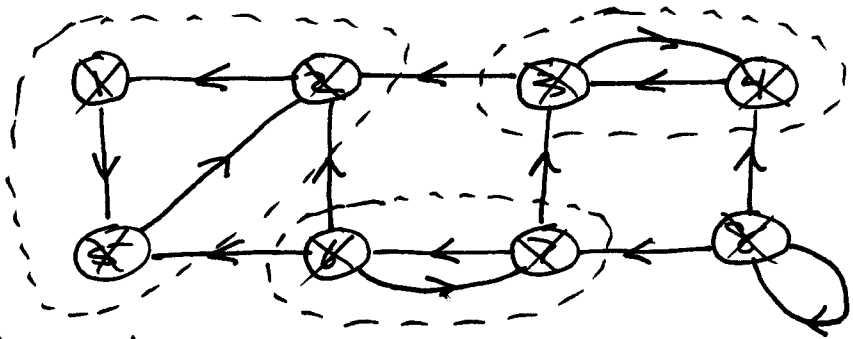
stack

1 ←
2 ←
5 ←
3 ←
7 ←
6 ←
4 ←
8

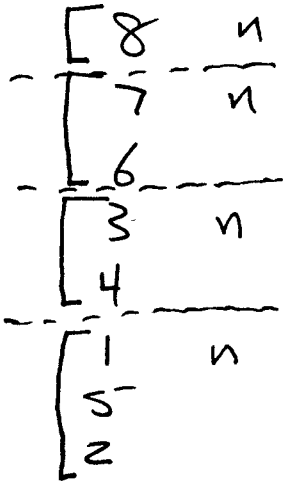
DFS forest



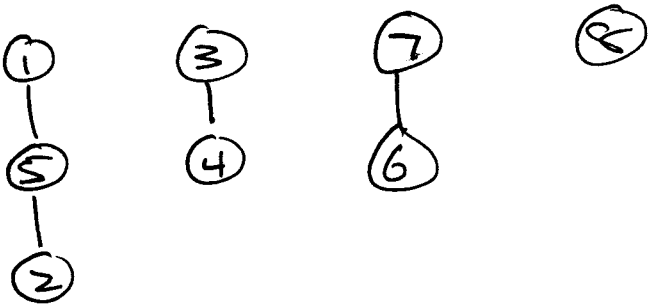
G^T



stack P



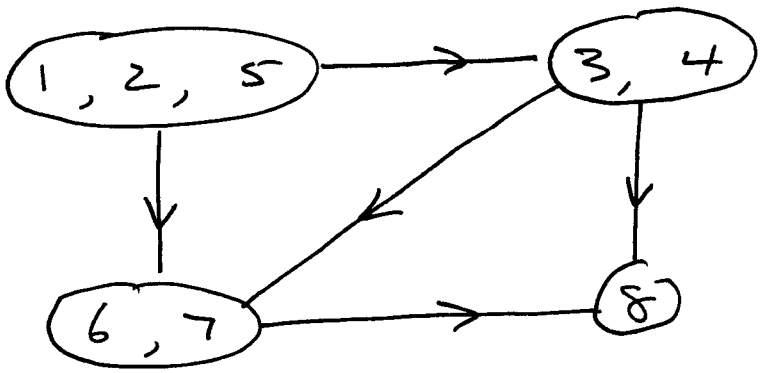
DFS forest



Component Graph (condensation graph) of G, denoted G^{scc}

- each scc of G is a vertex of G^{scc} : $\{C_1, C_2, \dots\}$
- edge $(C_i, C_j) \in E(G^{scc})$ iff there exists $x \in C_i$, $y \in C_j$ s.t. $(x, y) \in E(G)$.

example : G^{scc}



note: acyclic.

Top sort :