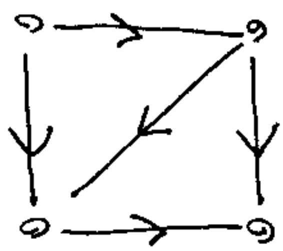
CSE 101 4-30-20

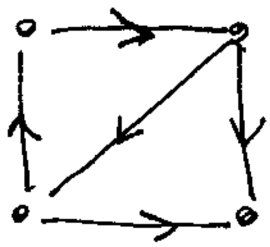
23.4 Topological sort { all digraphs now

Defn a digraph is called *acyclic* iff it contains no directed cycles.

Ex



acyclic



not acyclic

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

DAG ~ Directed Acyclic Graph

Defn

A topological sort of $V = V(G)$,

where G is a DAG, is a linear

ordering of V such that

if $(x, y) \in E(G)$

then x is before y in the ordering

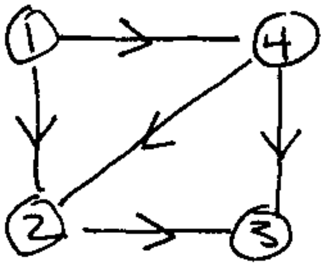
i.e. if we place vertices on a

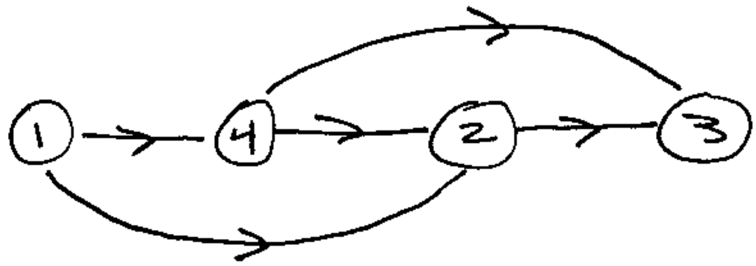
horizontal line, all edges go

left to right.

Ex.

G — Topological Sort of V(G)





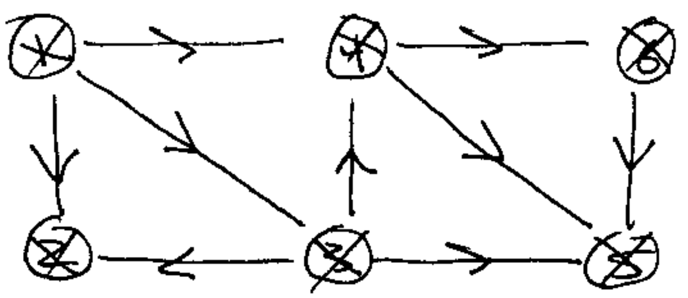
Note: A digraph admits a top-sort iff it is a DAG.

To find a top-sort of a DAG:

- call DFS(G), as vertices finish, push them onto a stack.
- when DFS is done, stack contains a top-sort of vertices.

Thus this algorithm works.

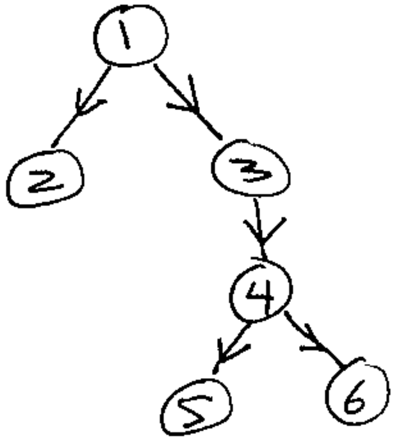
Ex. G



forest

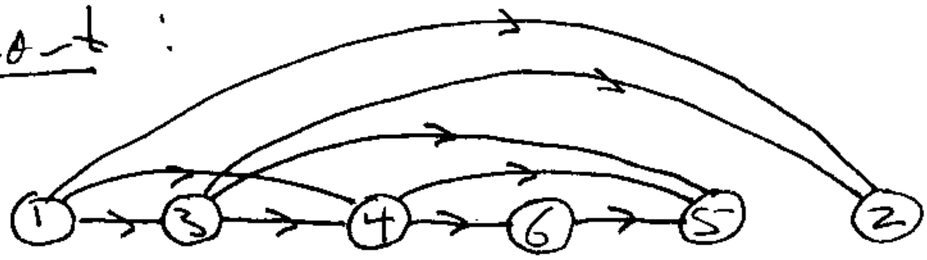
stack

1
3
4
6
5
2



x	d[x]	f[x]
1	1	12
2	2	3
3	4	11
4	5	10
5	6	7
6	8	9

Top sort:



Exercise: Find 3 other topological sorts in G.

Exercise

Figure out how to use DFS(G) to sort an array.

$$\text{array} \xrightarrow{\text{build}} \text{digraph} \xrightarrow{\text{DFS}} \text{top. sort}$$

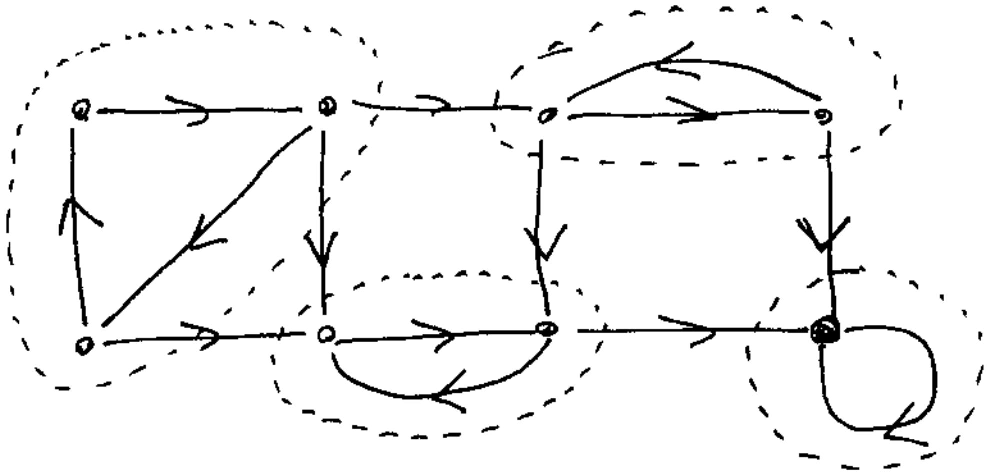
23.5 Strongly Conn. Components (SCC)

Recall: $U \subseteq V(G)$ is a SCC of G iff

(1) U is strongly connected

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

Ex.



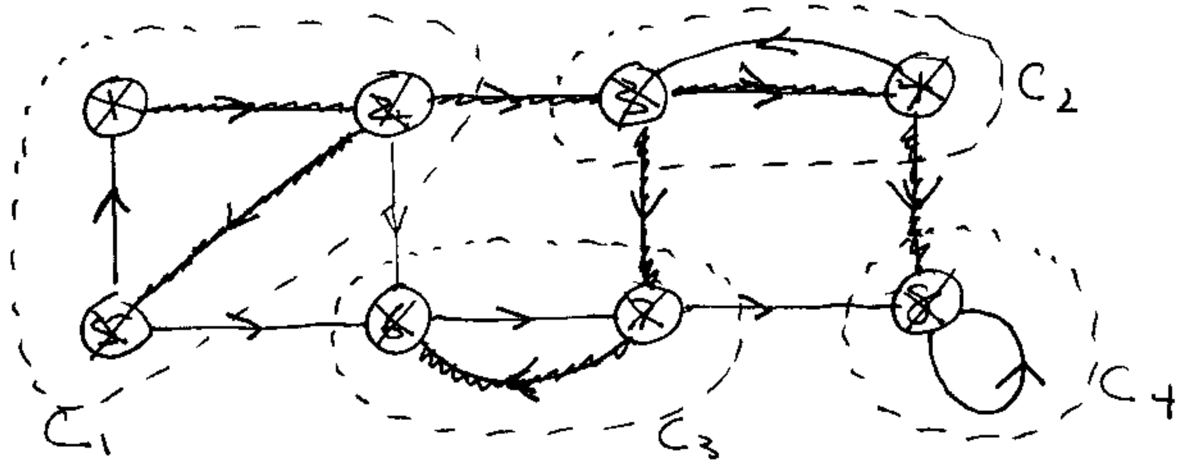
To find SCC's of G, Do:

- call DFS(G), as vertices finish, push them onto a stack
- compute G^T, i.e. reverse all directed edges in G.
- call DFS(G^T), process vertices in main loop of DFS by decreasing finish time from 1st call, i.e. pop off stack.

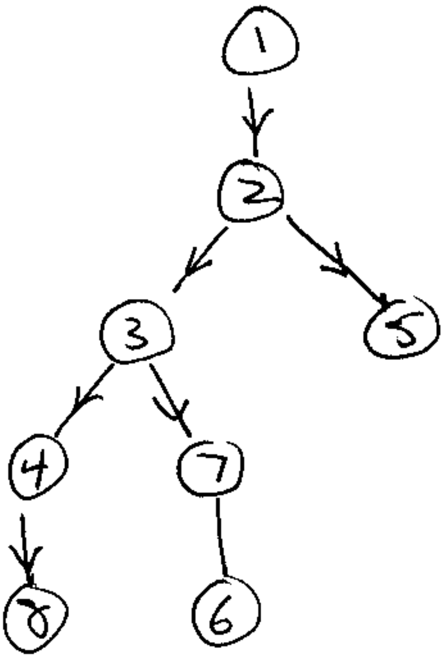
Thm
when done, trees in DFS forest from 2^{nd} call span the SCC's of G (and G^T).

Note
G and G^T have same SCCs

Ex
G



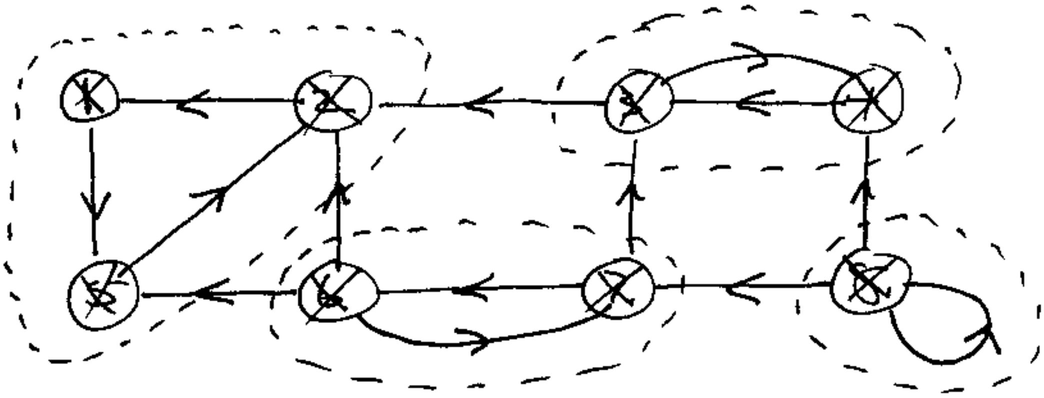
Forest



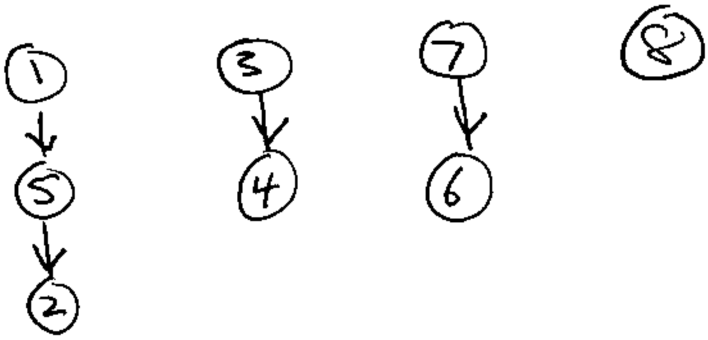
Stack

1
2
5
3
7
6
4
8

G^T



Forest



Stack

C_4 [8 n

C_3 [7 n, 6

C_2 [3 n, 4

C_1 [1 n, 5, 2

Defn

The component Graph (also condensation Graph) of a digraph G is the digraph G^{scc} in which

$$V(G^{scc}) = \{ \text{SCC's of } G \}$$

$$E(G^{scc}) = \{ (C_i, C_j) \mid \exists\, x \in C_i, \exists\, y \in C_j \text{ s.t. } (x,y) \in E(G) \}$$

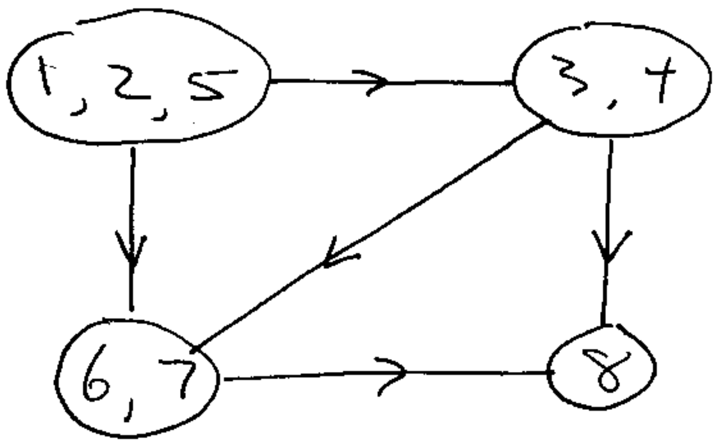
G^{SCC} has vertices:

$C_1 = \{1, 2, 5\}$

$C_2 = \{3, 4\}$

$C_3 = \{6, 7\}$

$C_4 = \{8\}$



Note: G^{SCC} is necessarily acyclic, i.e. a DAG, so admits a topological sort.

To extract a top-sort of G^{SCC}, look at state of stack after 2^{nd} call to $DFS(G^T)$

<u>output file</u>:

1: 1 5 2

2: 3 4

3: 7 6

4: 8

<u>note</u>:

$$(G^{scc})^T = (G^T)^{scc}$$

so in Pa4:

S: 1, 2, ..., n

DFS(G, S)

↓ dec. finish times

DFS(G^T, S)

↓ use state of S to obtain top-sort of G^{SCC}

$S \quad \underbrace{x \quad x \quad x \quad x \cdots \underline{x}}_{\text{control list for main loop}} \;\Big|\; \underbrace{y \quad y \quad y \cdots y}_{\text{stack}}$