CSE 101-02 4-14-22

Pa2: ~~Due Fri~~. ext. 1 last day.

Pa3: Posted

What structural information on a digraph G does DFS(G) give us?

Edge classification

- Tree: belong to Predecessor subgraph
- Back: Join Descendant → ancestor
- Forward: Join ancestor → descendant
- Cross: all else:
  - cousin → cousin
  - tree → tree

Prev. example:

<u>Tree</u>: (1,2), (2,5), (2,6), (6,7), (3,4), (3,8)

<u>Back</u> :

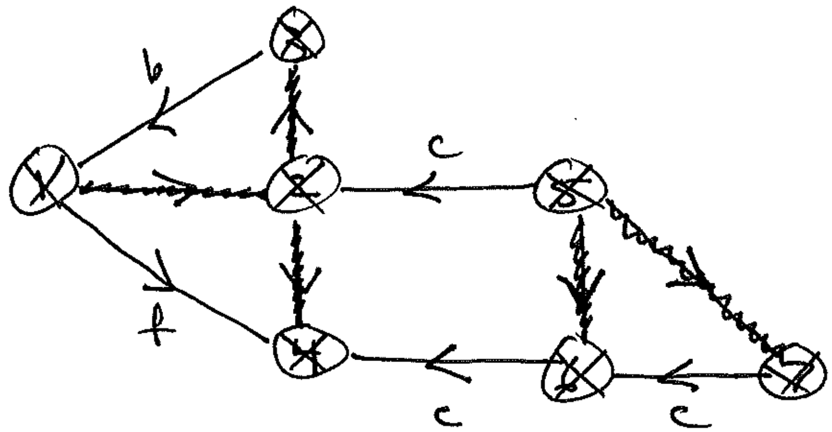
<u>Forward</u>: (1,5)

<u>Cross</u> : (6,5) , (8,4)
(3,2), (3,6) , (3,7), (8,7)

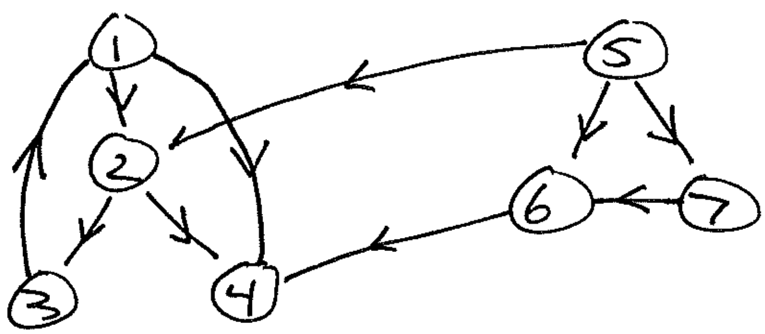
<u>Exercise</u>

Alter DFS so it prints out edge classification as it goes.

Ex.



Forest



Defn

A digraph is called Acyclic iff it contains no directed cycles.

Notation: DAG (directed acyclic graph)

Lemma

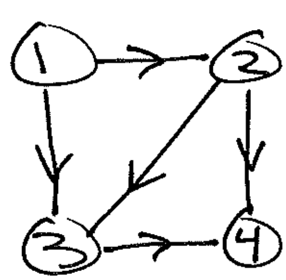
a digraph G is acyclic iff DFS(G) yields no back edges.

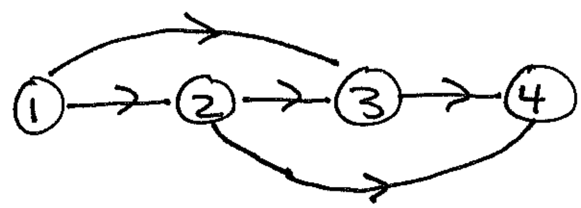
Defn

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 ordering)

Ex.

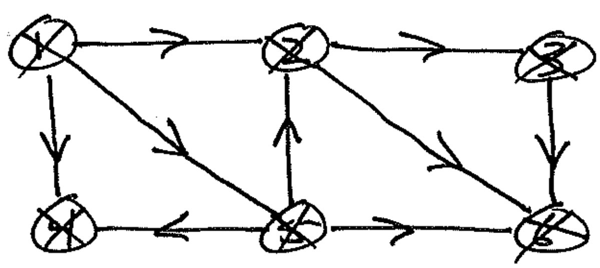




To find a T.S. of a DAG:

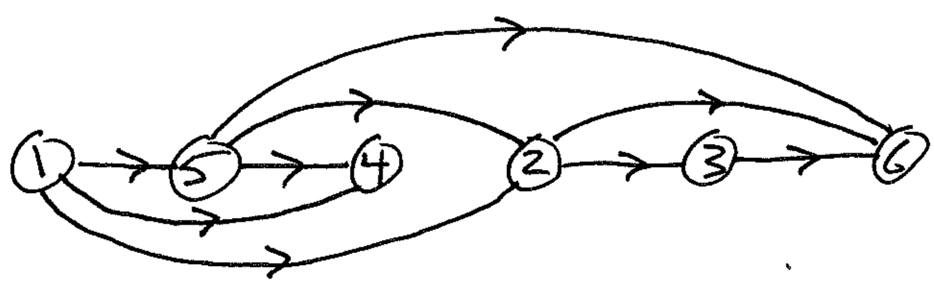
- run DFS(G), as vertices finish push them onto a stack: S
- when complete, S is the T.S. (top to bottom).

EX



stack

1
5
4
2
3
6



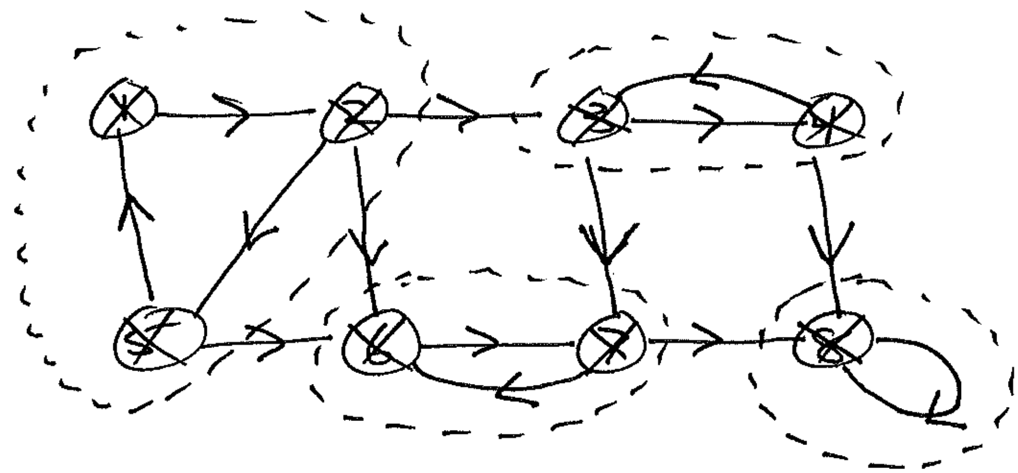
To find the S.C.C.s of a digraph (not necessarily a DAG)

- Run DFS(G), as vertices finish push onto a stack S
- compute the <u>transpose</u> G^T of G (i.e. reverse all dir. edges.)
- Run DFS(G^T), execute the main loop by decreasing finish times from 1st call, i.e. pop off the stack S.

When complete, the trees in the DFS forest have vertex set constituting the S.C.C.s of G.

Ex.

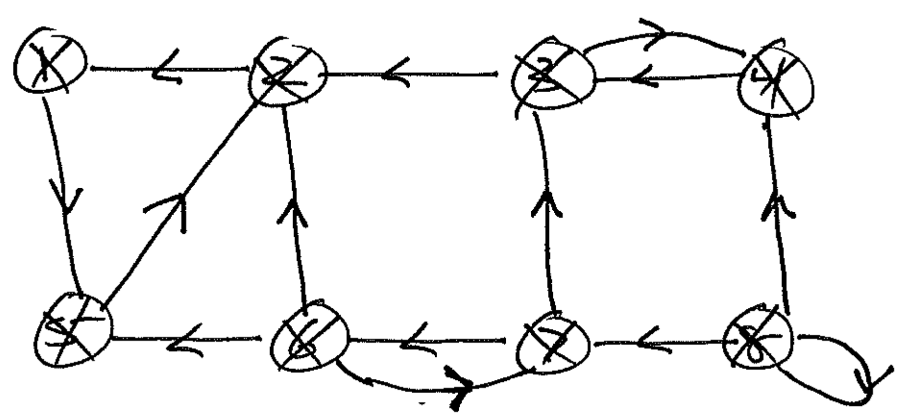
G



S

1 ✓
2 ✓
5 ✓
3 ✓
7 ✓
6 ✓
4 ✓
8 ✓

G^T



S

8

7

6

3

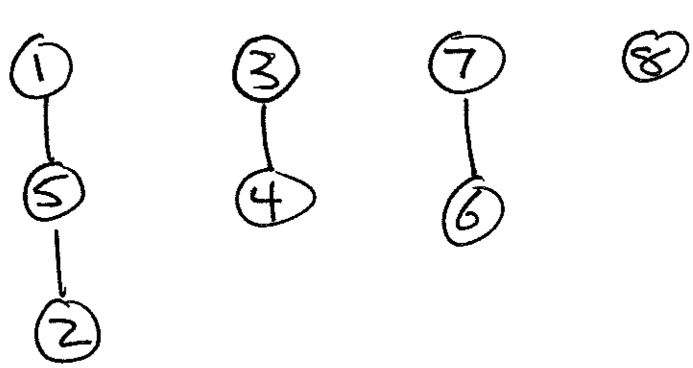
4

1

5

2

forest

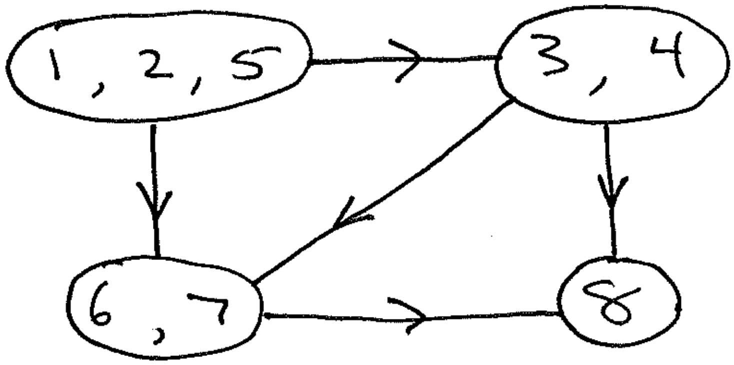


Defn

The component Graph (or condensation Graph) of G, denoted G^{SCC} is the digraph with

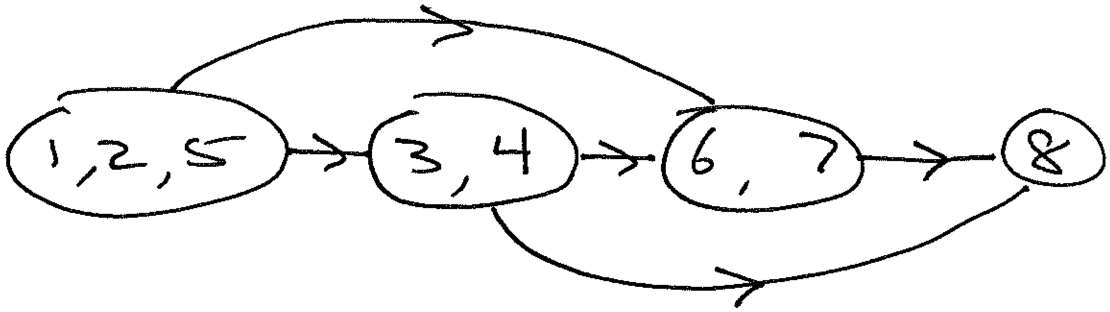
- Vertices: the S.C.C.s of G
- Edges: $(C_i, C_j) \in E(G^{SCC})$ iff. $\exists x \in C_i, \exists y \in C_j$ s.t. $(x, y) \in E(G)$.

Ex.



acyclic !!

Topological sort of G^{scc}



Observe:

- G & G^T have same S.C.C.s

and

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

To find a topological sort of G^{scc}, read 2^{nd} stack from bottom to top.

each nil parent in that list separates two S.C.C.s of G_T in T.S. order

C_4	8
C_3:	7
	6
C_2:	3
	4
C_1:	1
	5
	2

output file:

1:	1 5 2
2:	3 4
3:	7 6
4:	8

Pa3:

list S: 1, 2, 3 ...., n

↓

DFS(G, S)

↓ S: dec. finish times

DFS(G^T, S)

↓ S: dec. finish times

use order to

1. determine SCCs
2. determine T.S. of G^{SCC}