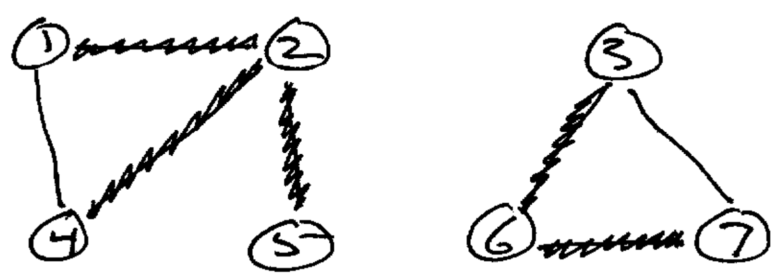
CSE 101 1-20-22

- back to in person Mon. Jan 31.
- pa2: Due ~~Fri.~~ Sat. (last ext.)
- pa3: posted
- graph algorithms handout (new version)

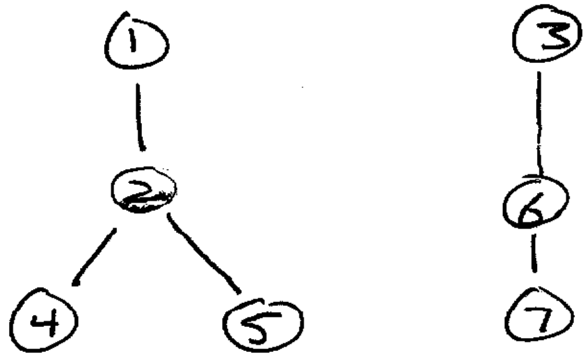
Ex.

G



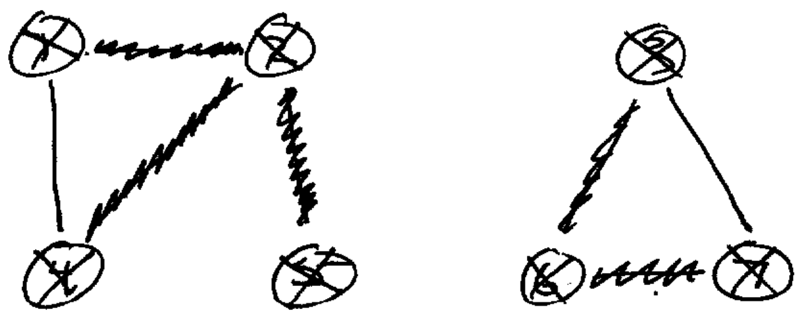
		adj	d	f	p
✓	1	2 4	1	8	n
✓	2	1 4 5	2	7	1
✓	3	6 7	9	14	n.
✓	4	1 2	3	4	2
✓	5	2	5	6	2
✓	6	3 7	10	13	3
✓	7	3 6	11	12	6

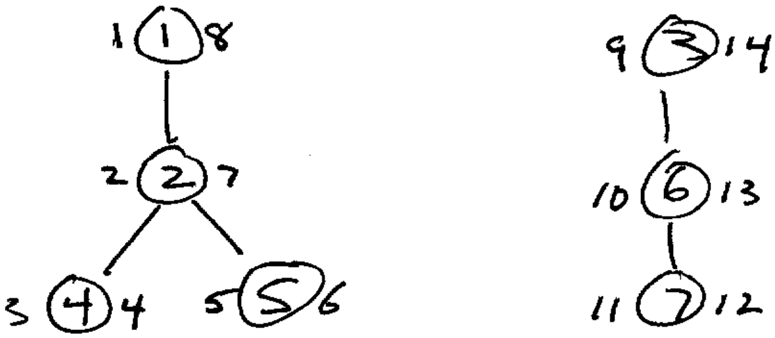
DFS forest



shortcut method:

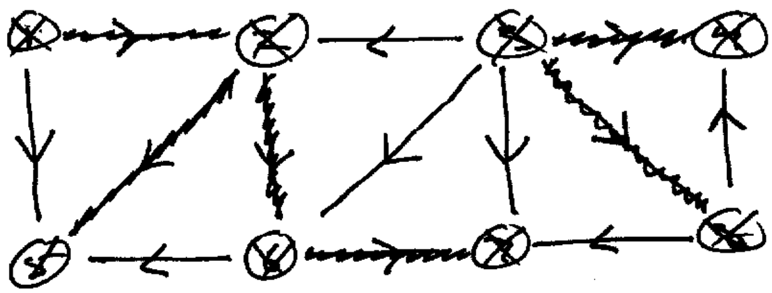
G



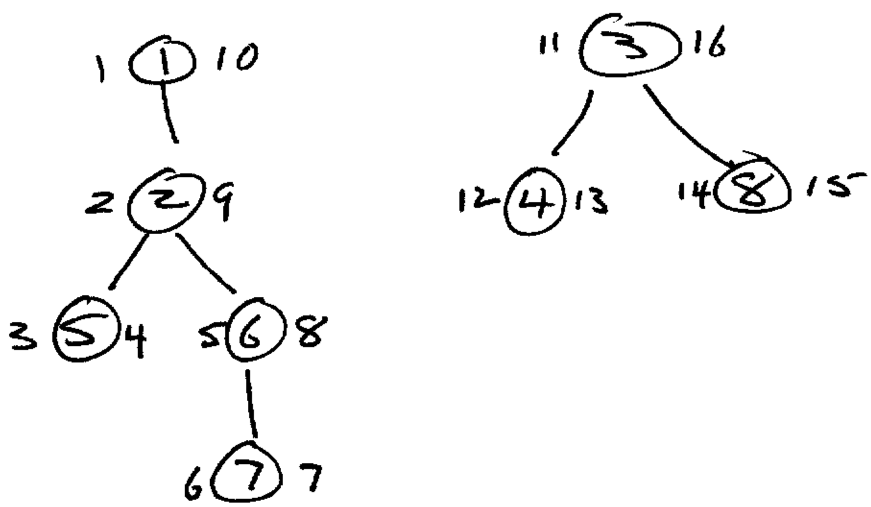


Ex. digraph

G

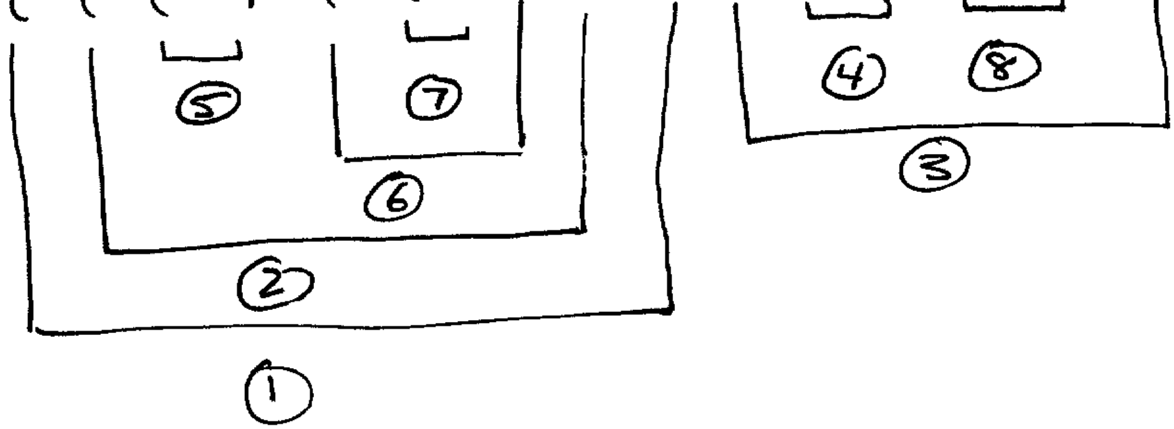


DFS Forest:



Parenthesis string

time:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
string:	(	(	(	)	(	(	)	)	)	)	(	(	)	(	)	)



Theorem (Parenthesis)

Suppose $x, y \in V(G)$ and $d[x] < d[y]$.
Then exactly one of the following hold.

(1) $d[x] < f[x] < d[y] < f[y]$ — neither x nor y a desc. of other

$$\underbrace{(\text{--------})}_{\text{gray } x} \qquad \underbrace{(\text{------})}_{\text{gray } y}$$

or

(2) $d[x] < d[y] < f[y] < f[x]$ — y is a desc. of x

$$(\text{----}\underbrace{(\text{------})}_{\text{gray } y}\text{----})$$

Proof see CLRS P.606.

Theorem (White Path)

Let $x, y \in V(G)$. Then y is a descendant of x (i.e. (2) holds) if and only if at the time $d[x]$ G contains an x-y path consisting of all white vertices.

Edge classification

- tree edges: belong to DFS forest
- back edges: join vertex to ancestor
- forward edges: join ancestor to descendant
- cross edges: every other edge
  - cousin to cousin
  - tree to tree.

Ex. (last example)

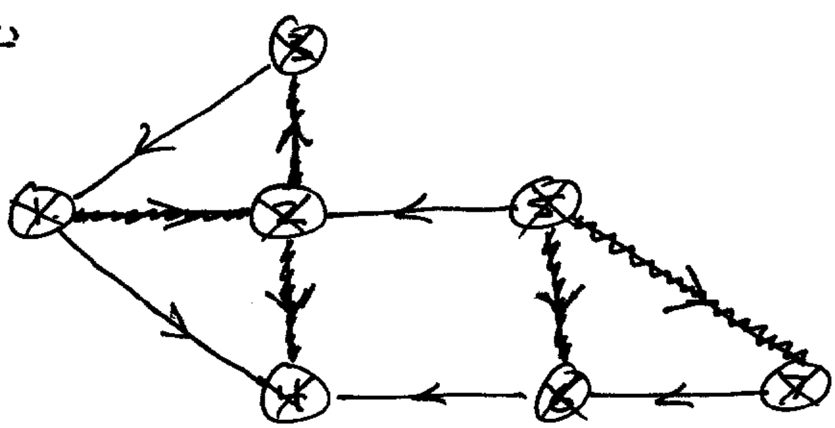
tree : (1,2), (2,5), (2,6), (6,7), (3,4), (3,8)

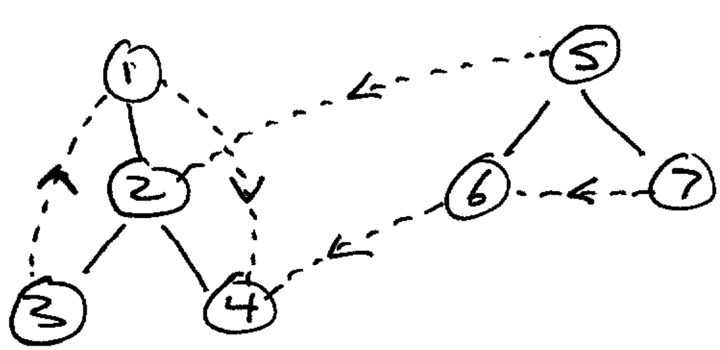
back :

forward: (1,5)

cross: $\underbrace{(6,5),\ (8,4)}_{\text{cousin to cousin}}\ \underbrace{(3,2),\ (3,6),\ (3,7),\ (8,7)}_{\text{tree to tree}}$

Ex.





tree:
(1,2), (2,3), (2,4)
(5,6), (5,7)

back: (3,1)

forward: (1,4)

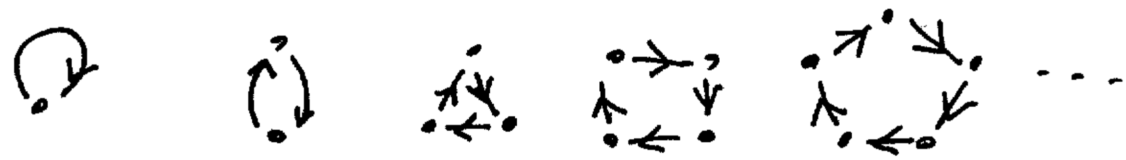
cross:

(7,6) cous-cous.

$\left.\begin{array}{l}(5,2)\\(6,4)\end{array}\right\}$ tree-tree

Topological sort

from now on: $G = (V, E)$ is a digraph.

cycle: 

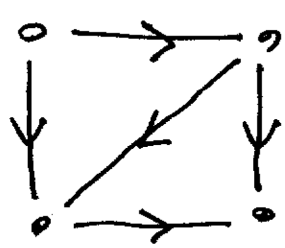
length 1 2 3 4 5 ...

Defn

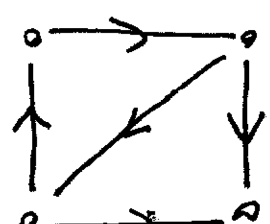
G is called acyclic iff it contains no directed cycles. (DAG)

directed acyclic graph

EX



acyclic
DAG



not acyclic

Lemma

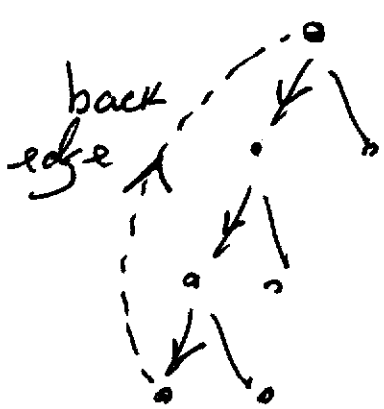
A digraph G is acyclic iff DFS(G) yields no back edges.

equivalently:

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

Proof: see handout
uses white Path theorem

idea



if have back edge, then have cycle.

Defn

let $G=(V,E)$ be a DAG.

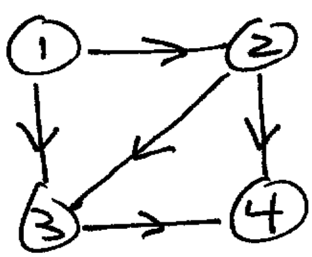
A <u>topological sort</u> of V is

a linear ordering of V

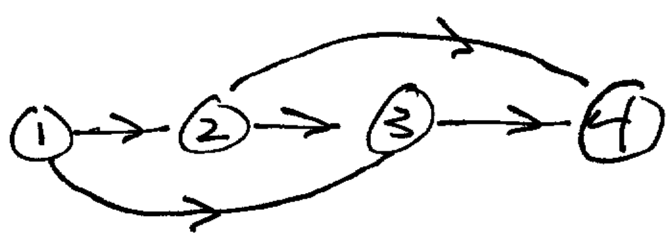
such that if $(x,y)\in E$, then

x is before y in the ordering.

Ex.

G 

DAG



all edges in
same direction