cse 101 1-21-21

- Quiz 2 : next Tuesday
- Pa3 : Posted
- Pa2 : ext 1 last day

DFS .... continue ....

Predecessor Subgraph :

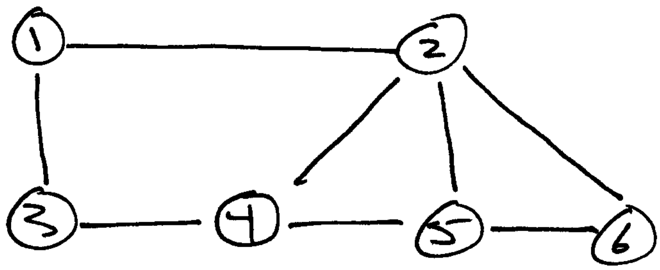
$$G_p = (V_p, E_p)$$

$$V_p = V(G)$$

$$E_p = \{ \underbrace{(p[x], x)} \mid p[x] \neq nil \}$$

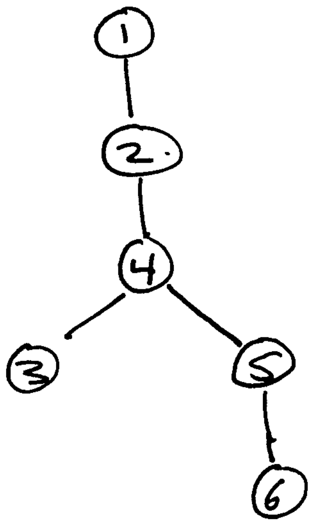
directed edge if G is a dir. graph
undirected edge if G a undir. graph

Ex.

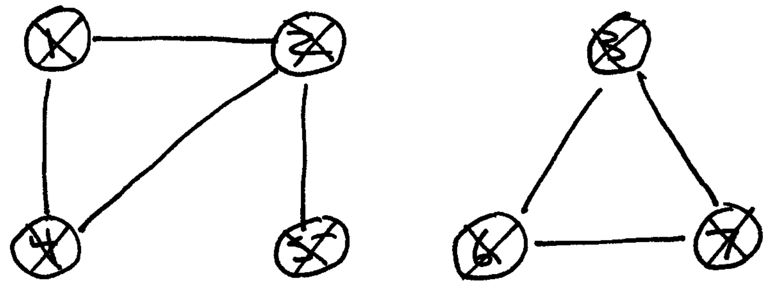


	adj	color	d	f	p
1	2 3	~~w~~ ~~g~~ b	1	12	N
2	1 4 5 6	~~w~~ ~~g~~ b	2	11	~~N~~ 1
3	1 4	~~w~~ ~~g~~ b	4	5	~~N~~ 4
4	2 3 5	~~w~~ ~~g~~ b	3	10	~~N~~ 2
5	2 4 6	~~w~~ ~~g~~ b	6	9	~~N~~ 4
6	2 5	~~w~~ ~~g~~ b	7	8	~~N~~ 5

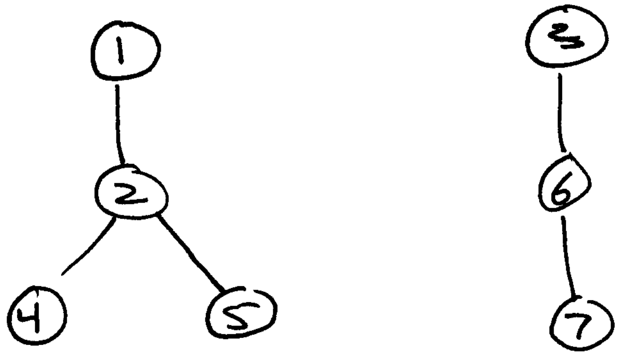
DFS forest



Ex.



DFS forest



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

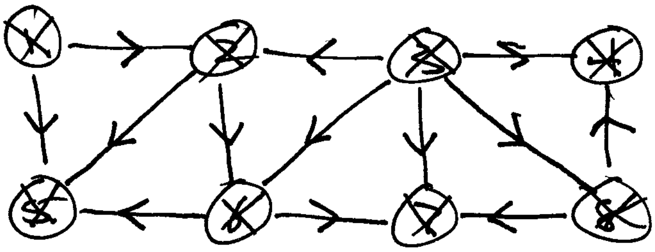
<u>Exercise</u> (G an <u>undir.</u> graph)
add an attribute <u>cc[x]</u> to each vertex x. Introduce a new variable <u>compNum</u> in DFS (local to DFS, static over all recursive calls to Visit( ), like time.)

figure out what to do with cc[x] and compNum such that after DFS(G), each $x \in V(G)$ has a cc-value and

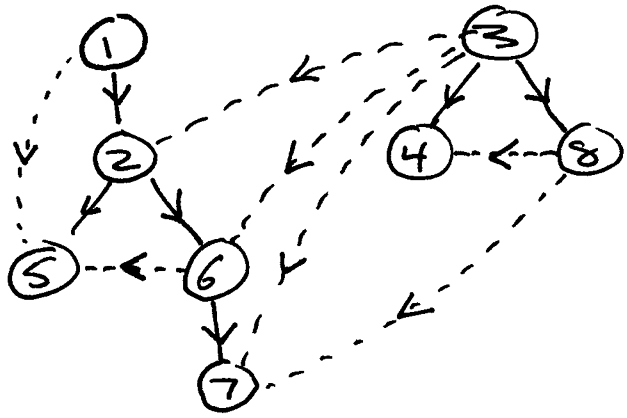
$$cc[x] == cc[y]$$

iff x, y lie in same connected component.

Ex.



DFS forest



Tree edges

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

Back edges ϕ

Forward edges

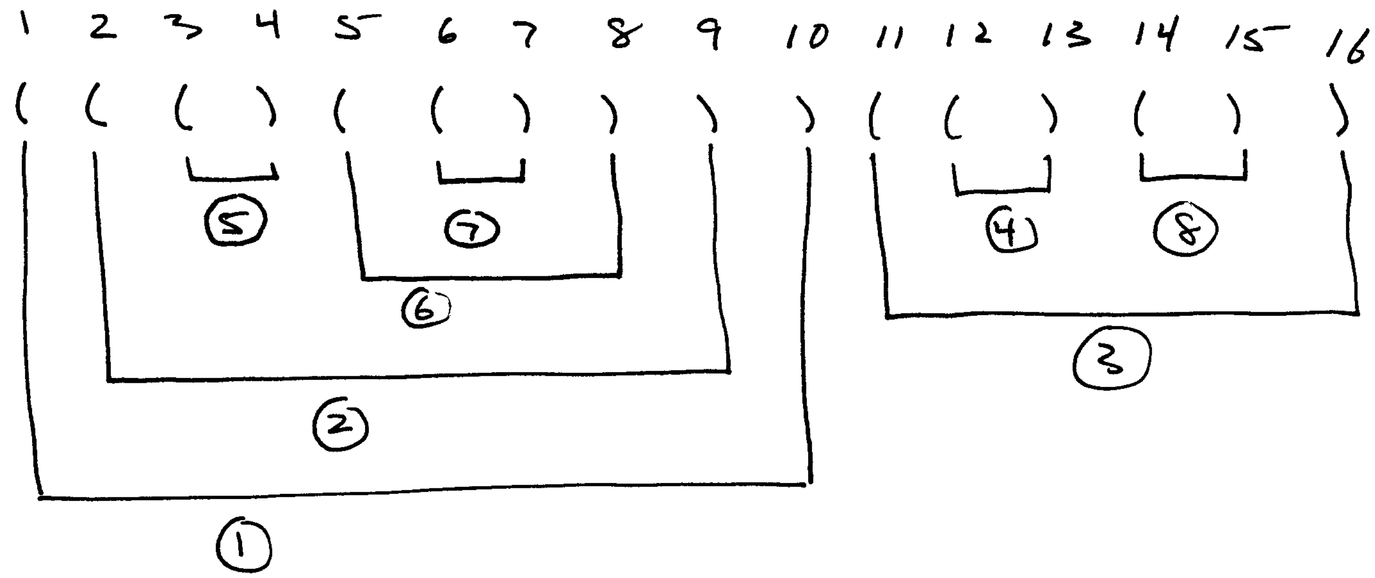
(1, 5)

Cross edges

(6, 5), (8, 4)
(3, 2), (3, 6), (3, 7), (8, 7)

	adj	d	f	p
1	2 5	1	10	n
2	5 6	2	9	1
3	2 4 6 7 8	11	16	n
4		12	13	3
5		3	4	2
6	5 7	5	8	2
7		6	7	6
8	4 7	14	15	3

Parenthesis string



Thm

Suppose $x, y \in V(G)$ and $d[x] < d[y]$. Then either

① $d[x] < f[x] < d[y] < f[y]$

( ) ( )

• y is not a descendant of x

or

② $d[x] < d[y] < f[y] < f[x]$

(( ))

$\underbrace{\qquad}_{y \text{ gray}}$

$\underbrace{\qquad\qquad}_{x \text{ gray}}$

• y disc. while x gray
• y is a descendant of x

Thm (white Path)

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

Edge classification

Run DFS. Then

- Tree edges: Belong to DFS forest G_p
- Back edges: join a vertex to one of its ancestors.
- Forward edges: join vertex to one of its descendants
- Cross edges: all else
  - tree to tree
  - cousin to cousin