CSE 101 4-13-21

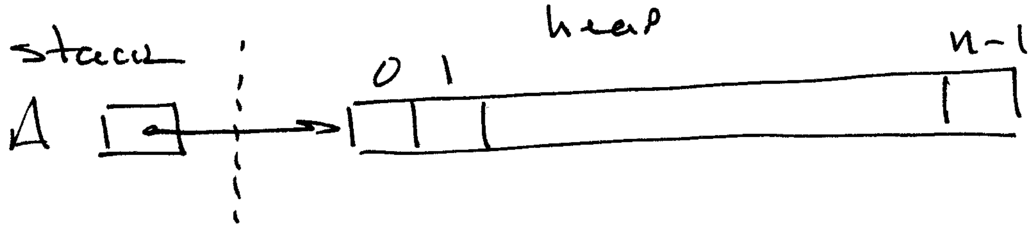
Pa1 & other assignments:

Dynamic allocation of arrays in C

<u>normal:</u>

variable

`int* A = calloc(n, sizeof(int))`



⋮

`free(A)`

<u>As of ISO C99</u>: Variable length arrays (VLA)

`int A[n];` can be a variable

limitation: cannot be used outside function in which it was defined.

Pa2 : ext 1 day

Depth First search (DFS)

what problem does DFS solve?

many structural questions are answered by DFS.

vertex attributes: let $x \in V(G)$

- color[x] : white, gray, black
- P[x] : predecessor of x
- d[x] : discover time
- f[x] : finish time

DFS(G) has no source.

Recursive subroutine: Visit(x)

time: local to DFS()
static over all recursive
calls to Visit()

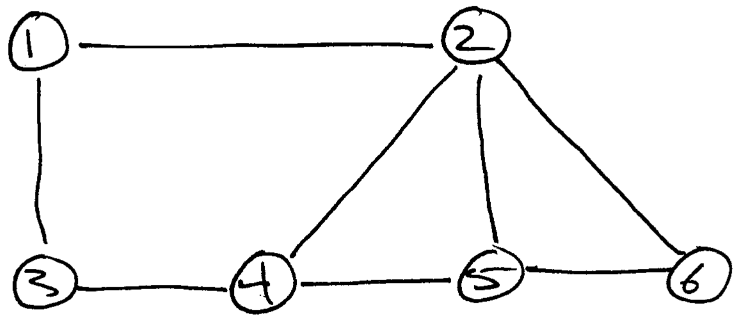
Predecessor subgraph: $G_p = (V_p, E_p)$

$$V_p = V(G)$$

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

ordered pair if digraph
unordered pair { } if graph.

Ex.

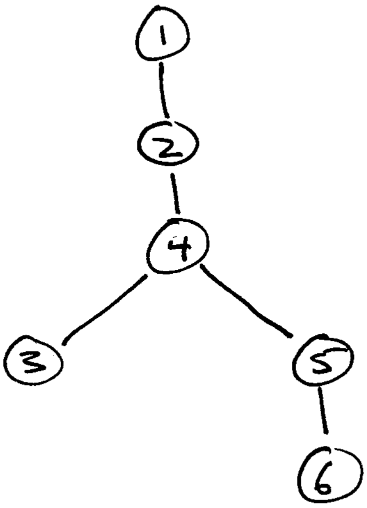


	adj	color	d	f	p
1	2 3	~~w~~ ~~g~~ b	1	12	n
2	1 4 5 6	~~w~~ ~~g~~ b	2	11	~~n~~ 1
3	1 4	~~w~~ ~~g~~ b	4	5	~~n~~ 4
4	2 3 5	~~w~~ ~~g~~ b	3	10	~~n~~ 2
5	2 4 6	~~w~~ ~~g~~ b	6	9	~~n~~ 4
6	2 5	~~w~~ ~~g~~ b	7	8	~~n~~ 5

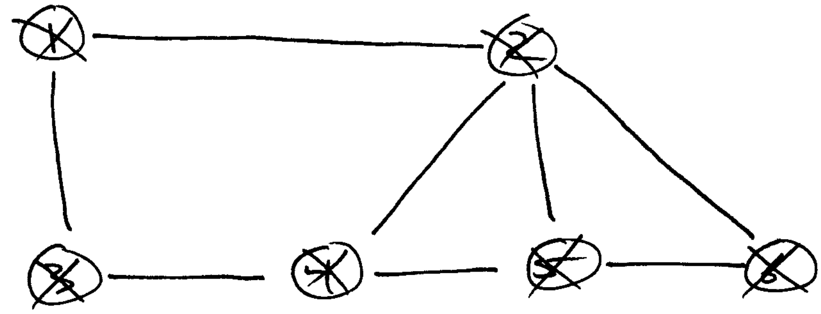
time

~~0~~
~~1~~
~~2~~
~~3~~
~~4~~
~~5~~
~~6~~
~~7~~
~~8~~
~~9~~
~~10~~
~~11~~
12

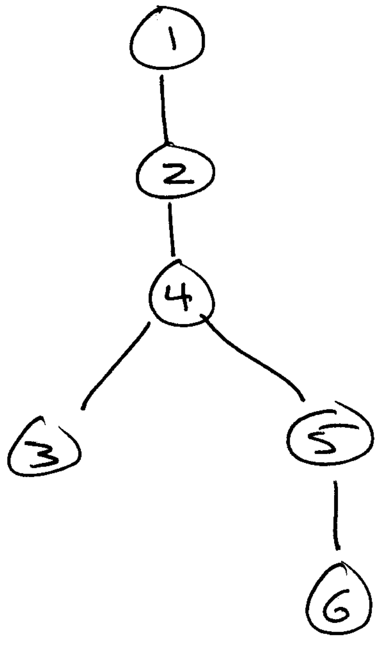
DFS forest



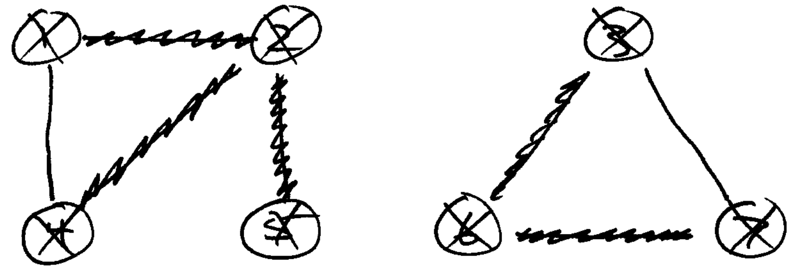
Again: Just look at graph



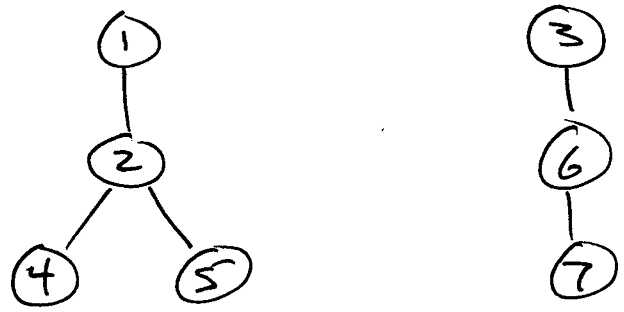
DFS Forest:



Ex.



Forest



Thm

In an undirected graph G, the # of trees in DFS forest is the number of connected components in G.

In fact the trees in DFS forest span the connected components

↑

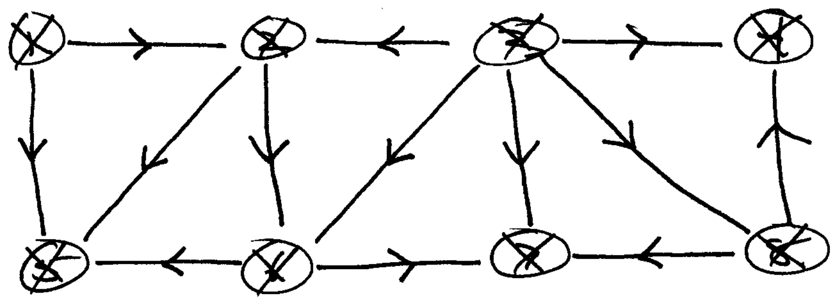
Cover all vertices in conn. comp.

Exercise

alter pseudo-code for DFS so as to place a 'component' stamp on each vertex identifying which conn. comp. it belongs to.

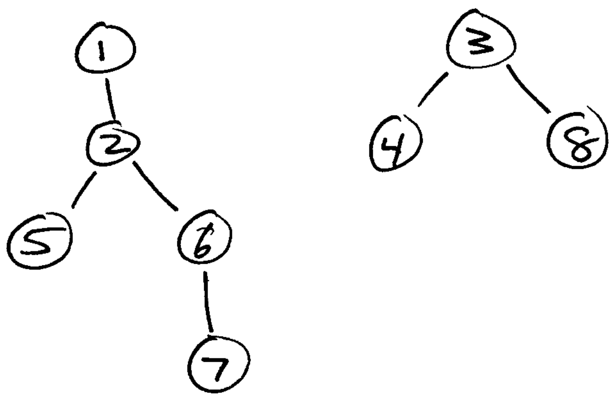
Hint. add another var to DFS, called count, like time, static over all recursive calls. increment every time Visit() is called from main loop of DFS. add new attribute to each vertex: comp[x].

Ex.



		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

DFS forest



In a digraph, there are 2 kinds of connectivity:

- <u>weak connectivity</u>: underlying undirected graph is connected

- <u>strong connectivity</u>:

  <u>Defn</u> A digraph $G = (V, E)$ is called strongly connected iff for all $x, y \in V$, both x is reachable from y and y is reachable from x.

<u>note</u>: y <u>reachable</u> from x means there exists a directed x-y path in G

$$\underset{x}{\circ} \to \circ \to \circ \cdots \circ \to \underset{y}{\circ}$$

more generally, a subset

$$S \subseteq V$$

is said to be strongly connected iff for all $x, y \in S$: x is reachable from y and y from x.

Defn
$S \subseteq V$ is called a strongly connected component (SCC) iff both

(1) S is strongly conn.
(2) S is maximal w.r.t. property (1)