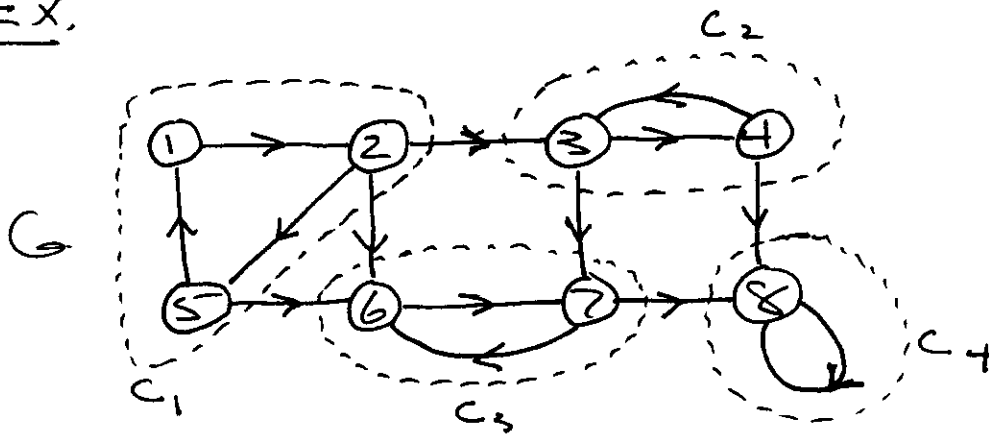


- Part : ext 2 more days
- Part handout : fix error (6,7) and (7,6).

finding SCC's in a digraph

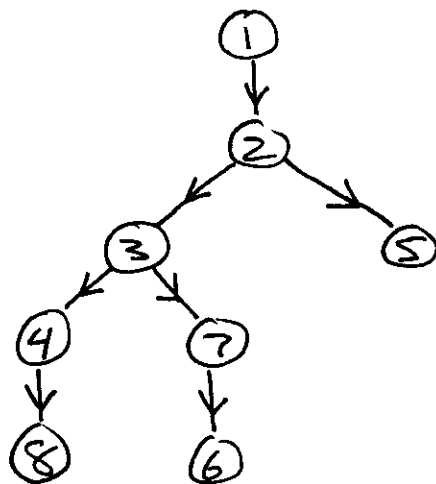
Ex.

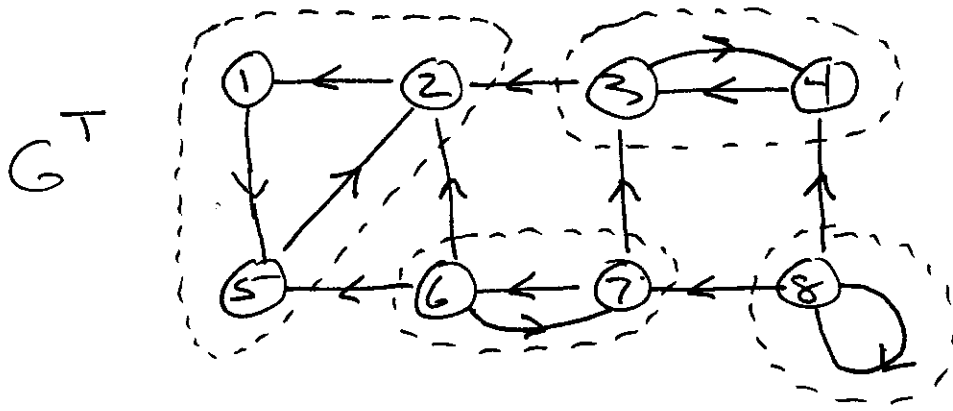


stack

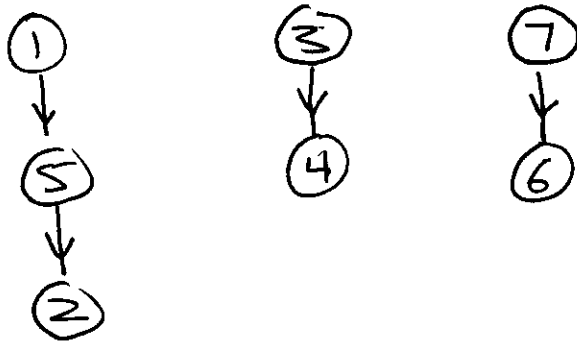
1 ✓
2 ✓
5 ✓
3 ✓
7 ✓
6 ✓
4 ✓
8 ✓

forest





forest



⑧

	stack	P
C_4	8	n
C_3	7	n
	6	7
C_2	3	n
	4	3
C_1	1	n
	5	1
	2	5

strong components of G :

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

$$C_2 = \{3, 4\}$$

$$C_3 = \{6, 7\}$$

$$C_4 = \{8\}$$

output at Pa 4

1: 1 5 2

2: 3 4

3: 7 6

4: 8

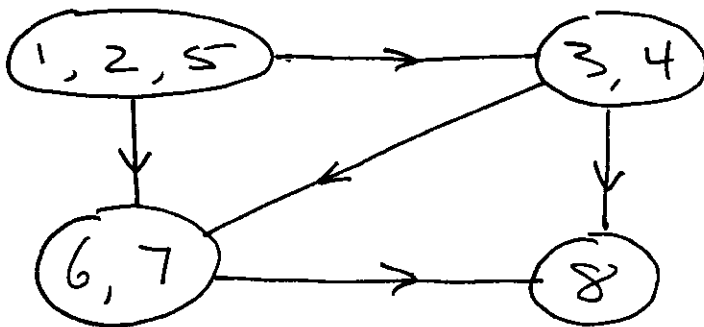
Defn

The component g -graph of a digraph G (also condensation g -graph), Denoted G^{scc} is the digraph with

$$V(G^{scc}) = \{ \text{strong components of } G \}$$

$$E(G^{scc}) = \{ (c_i, c_j) \mid \text{there exists } x \in c_i \text{ and } y \in c_j \text{ st. } (x, y) \in E(G) \}$$

Ex G^{scc}



notice:

G^{scc} is a DAG, so has a topological sort.

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

Dictionary ADT

state : a set of ordered Pairs
(key, value)

i.e. a state is a set:

$$\{(k_1, v_1), (k_2, v_2), (k_3, v_3), \dots\}$$

option : keys are distinct.

Operations

- $\text{lookup}(\text{key})$: return value assoc. with key, or nil if key does not exist.
- $\text{insert}(\text{key}, \text{value})$: adds new Pair to dictionary $\left[\begin{array}{l} \text{Pre: } \text{lookup}(\text{key}) = \text{nil} \\ \text{only if distinct keys} \end{array} \right]$
- $\text{delete}(\text{key})$: removes (key, value) from dictionary. $\left[\text{Pre: } \text{lookup}(\text{key}) \neq \text{nil} \right]$

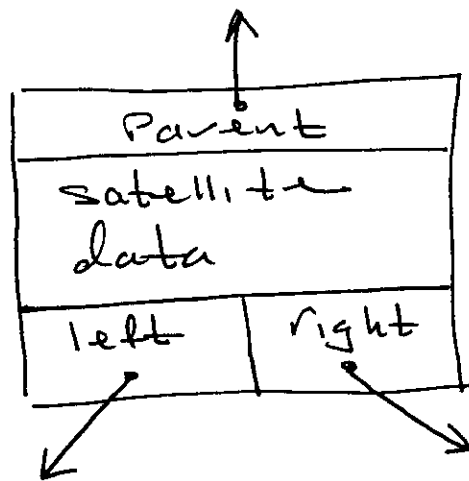
how to implement Dictionary ADT? 15

one way

Binary Search Tree (BST)

see chap 12 in CLRS.

Node
Object



Dictionary
(key, val)
[simplicity:
key only]

BST Properties:

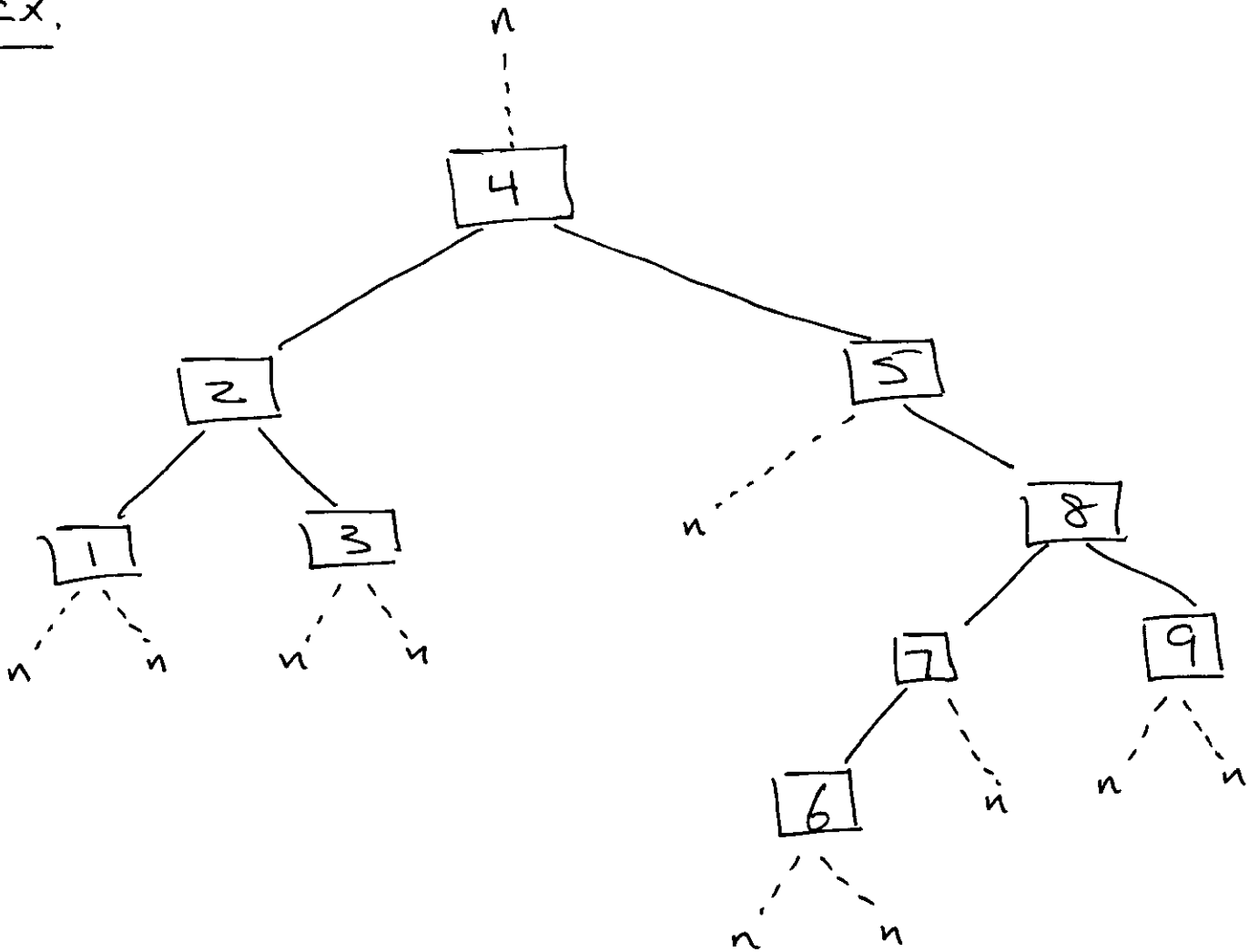
let x, y be nodes in a BST. Then

(1) if y is in left subtree of x ,

then $\text{key}[y] \leq \text{key}[x]$

(2) if y is in the right subtree of x ,

then $\text{key}[x] \leq \text{key}[y]$.

Ex.Defn

- A leaf is a node with no children
- an internal node is a non-leaf.

when implementing : recommend that we have a dummy node called NIL, as a field in DictionaryObj.