CSE 101-02 5-17-22

- mid 2 : this Thursday (5-19)
- PA6 : ext. one last day (Wed.)
- Dictionary ADT
- BSTs

runtimes : traversals : $\Theta(n)$

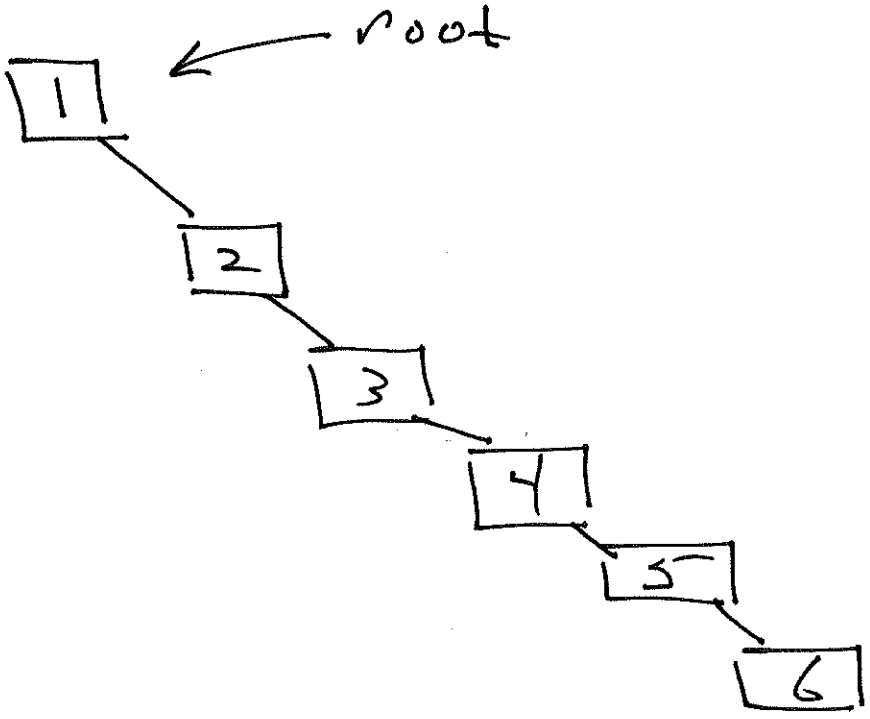
n = # nodes

Queries : TreeMin()
" Max()
" Search
Successor()
Predecessor()

runtime of Queries is

$\Theta(height(T))$

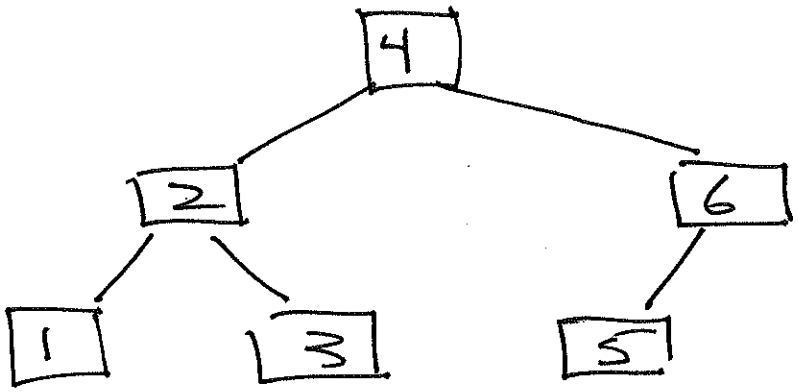
Ex. worst case for Queries



every node has only 1 child

cost of search is $n=6$.

Ex. best case for query



cost of search = height(T) = 2

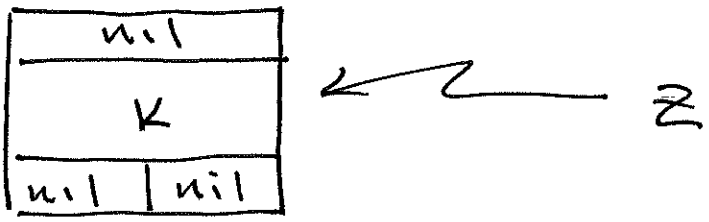
In general: for n nodes

$$\log n \leq \text{height}(T) \leq n$$

(12.3) Insertion

To insert new node z and maintain the BST Properties

- set key[z] = k
- set p[z] = left[z] = right[z] = nil

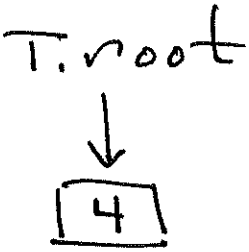


- find a node to adopt z as a child.

strategy: search for k in T
find the nil child where k would reside

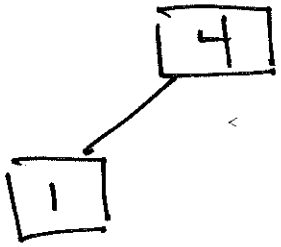
Ex. Keys: 4 1 7 2 8 3 5 6

initially T is empty, T.root = nil

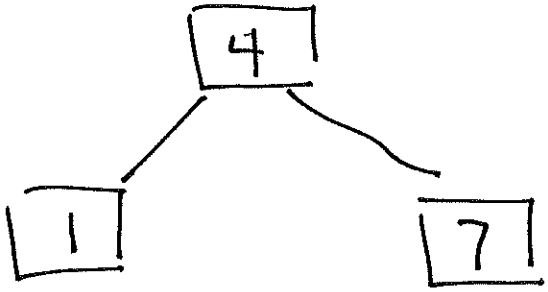


insert(4)

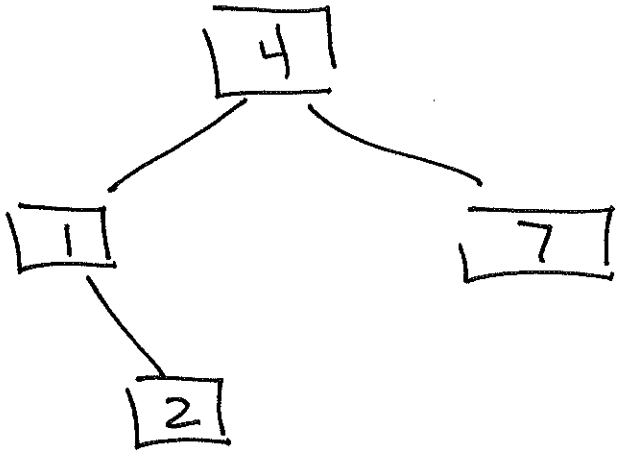
insert(1)



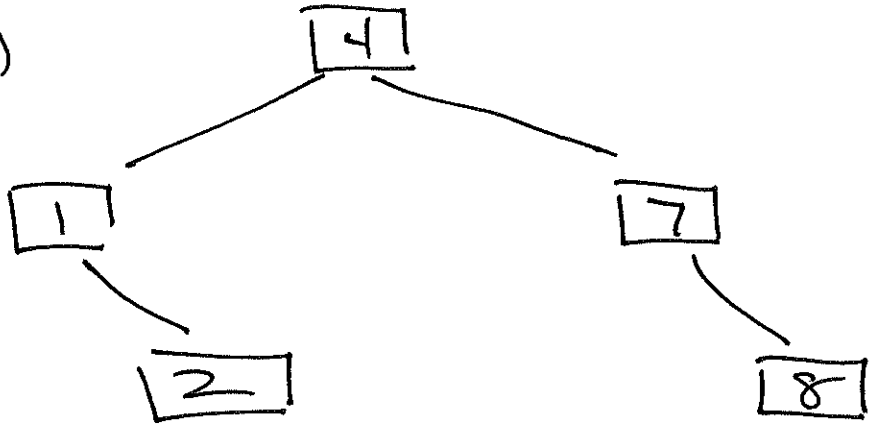
insert(7)



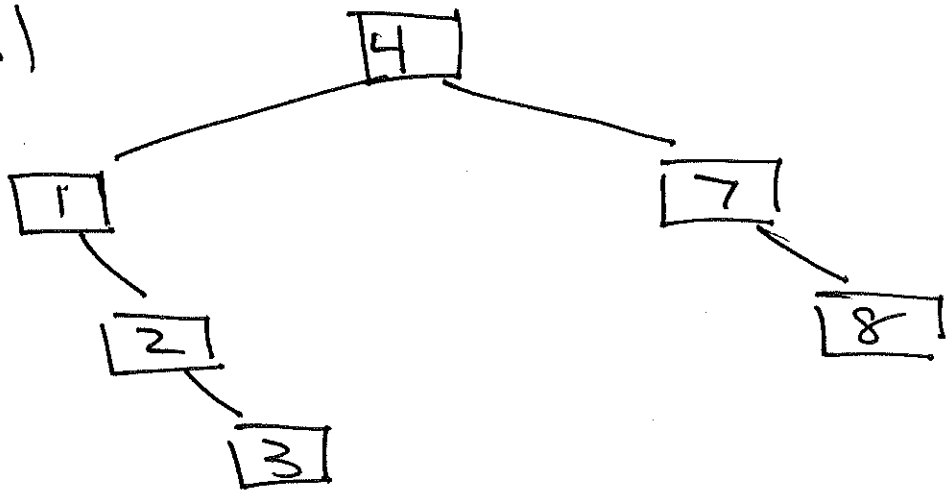
insert(2)



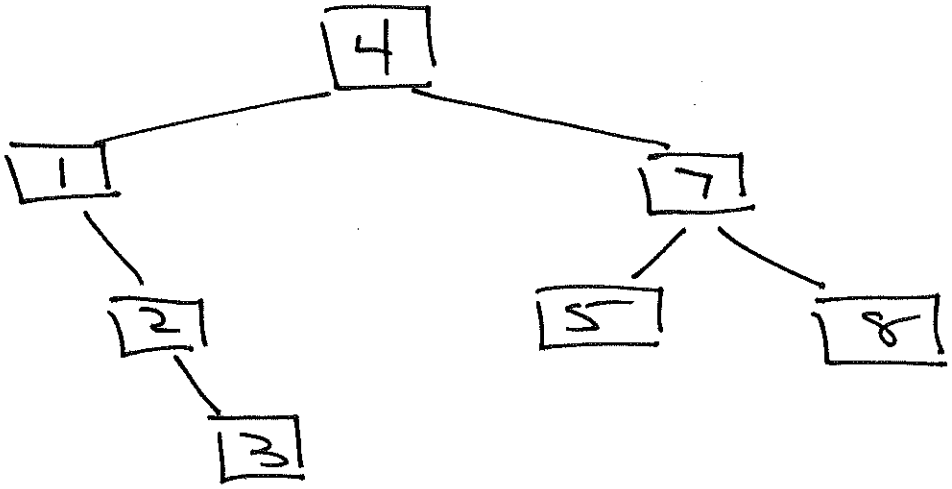
insert(8)



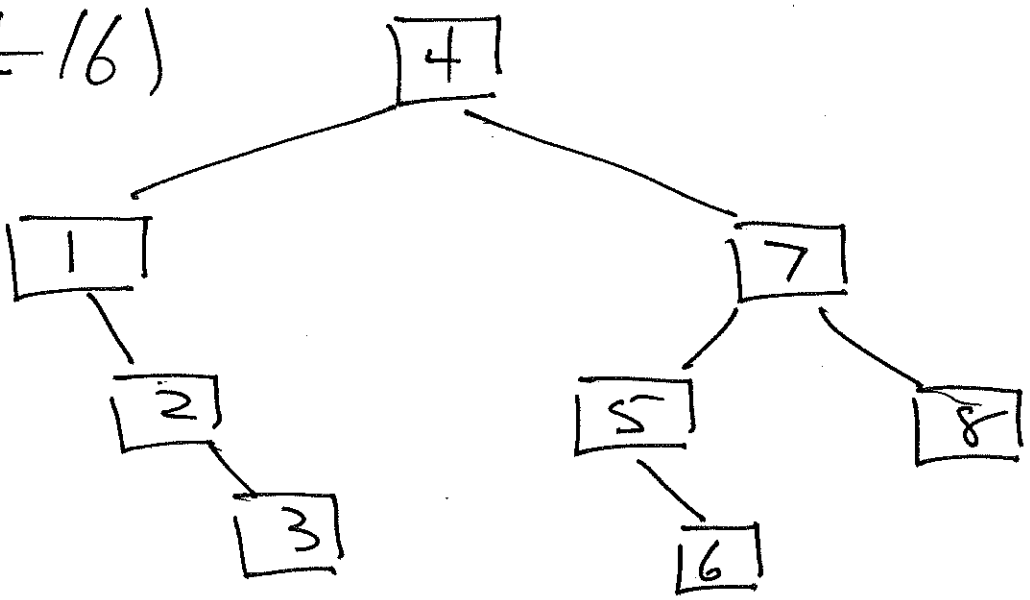
insert(3)



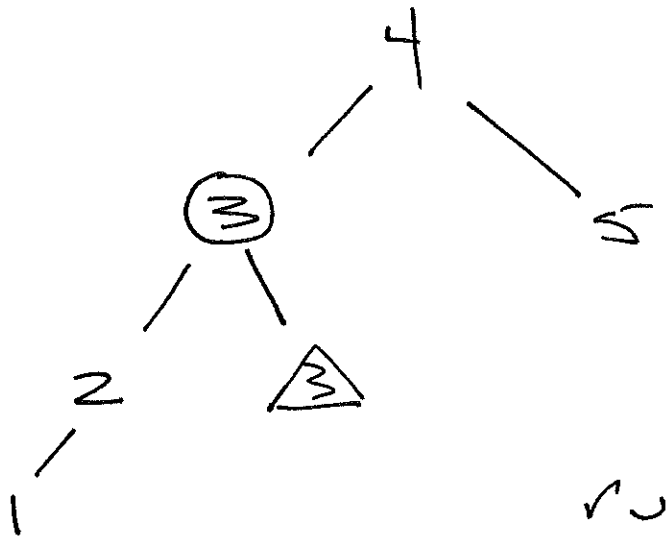
insert(5)



insert(6)



Ex. Keys: 4 3 2 1 3 5



runtime = $\Theta(\text{height}(T))$

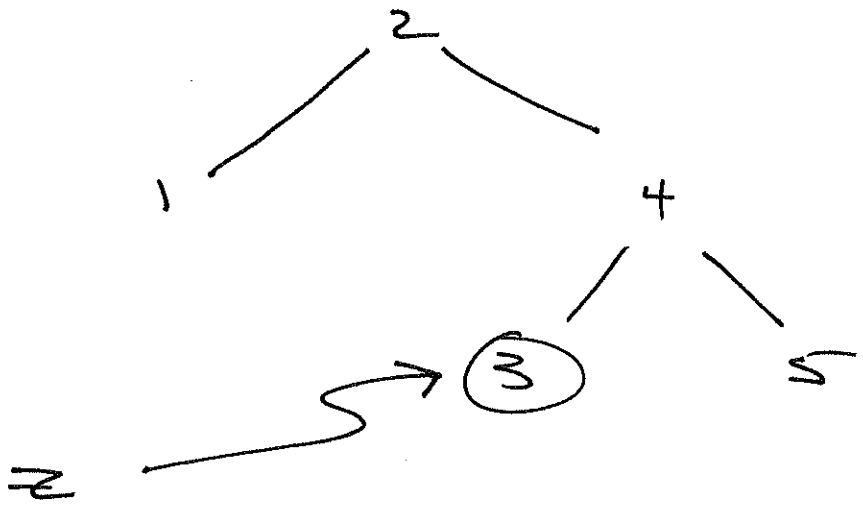
(12.4) Deletion

3 cases: to delete z

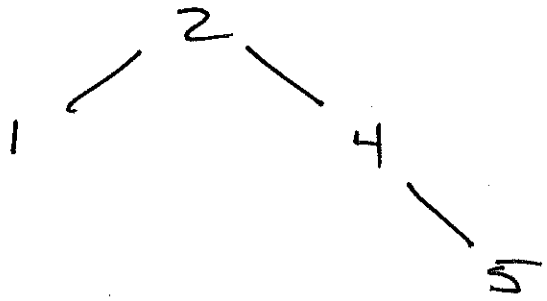
Case 1: z is a leaf (no children)

detach z from its parent.

Ex.



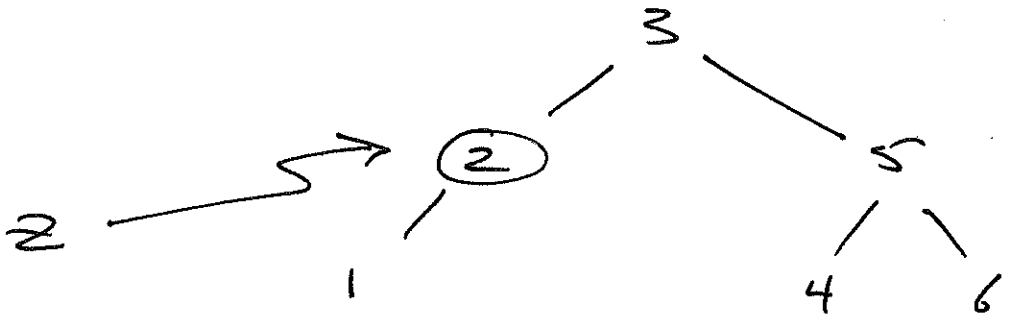
delete(3)



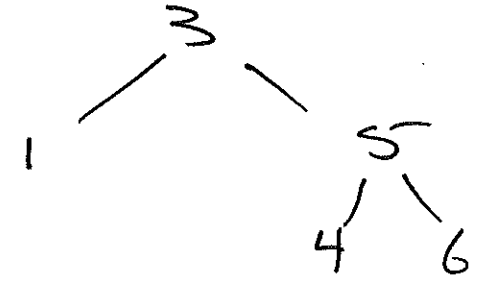
Case 2: z has 1 child

(2.1) z has right but no left

→(2.2) z has left but no right



delete(z) (case 2.2)



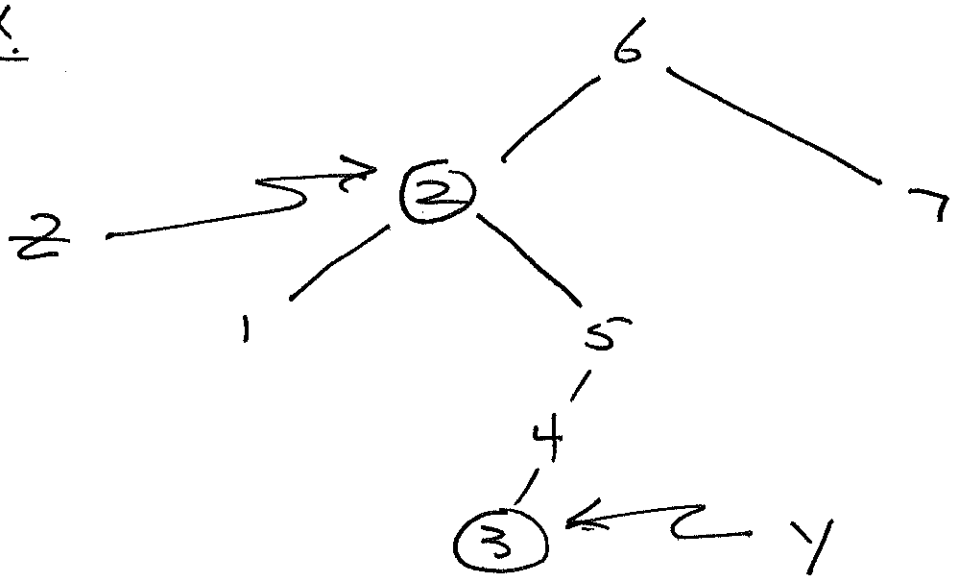
Exercise: draw a pic. of subcase 2.1.

Case 3: z has 2 children

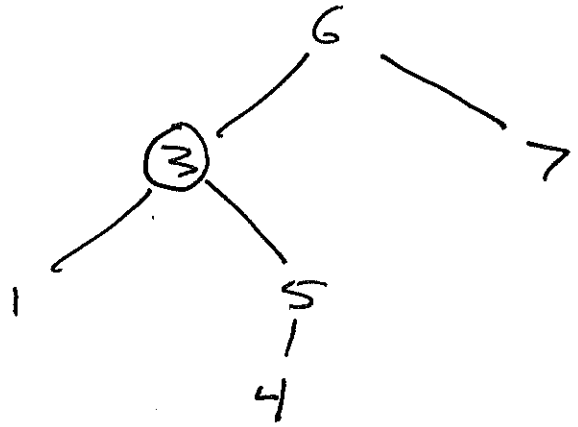
replace z by z's successor y

(which as no left child)

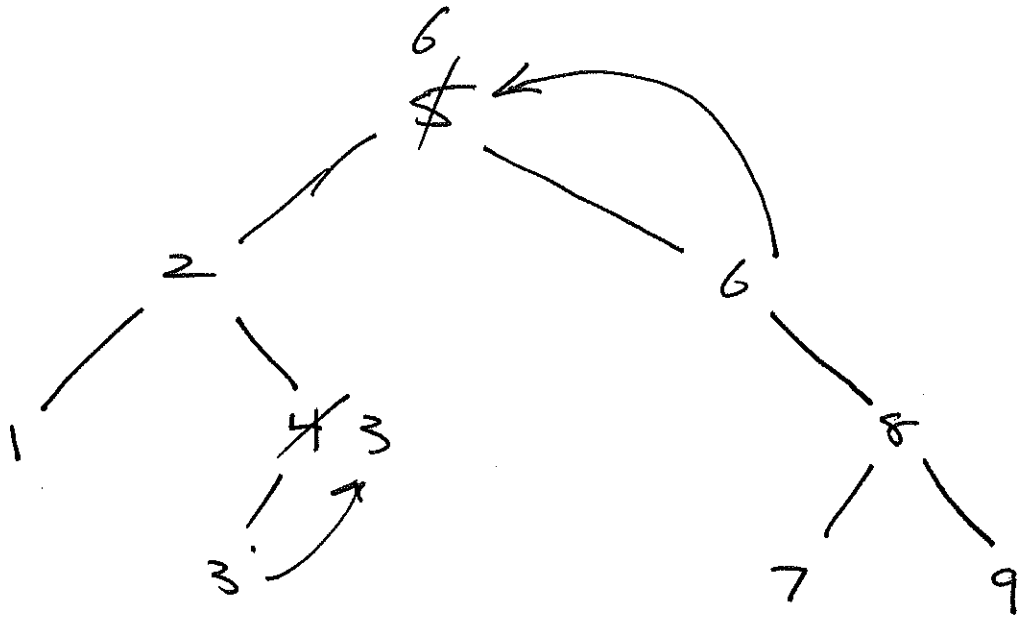
Ex.



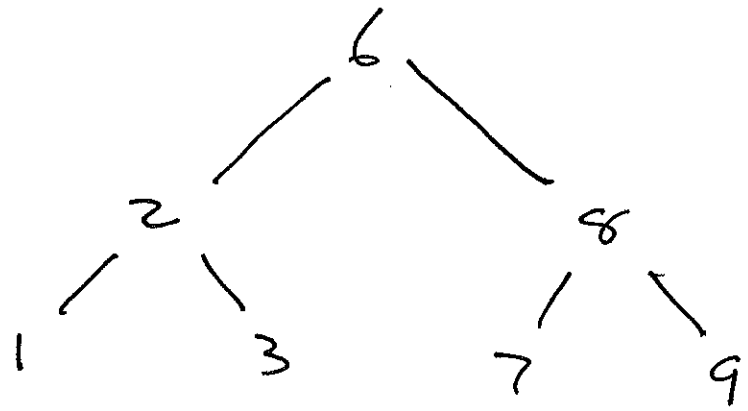
delete(2)



Ex insert: 5 2 6 4 3 8 7 1 9

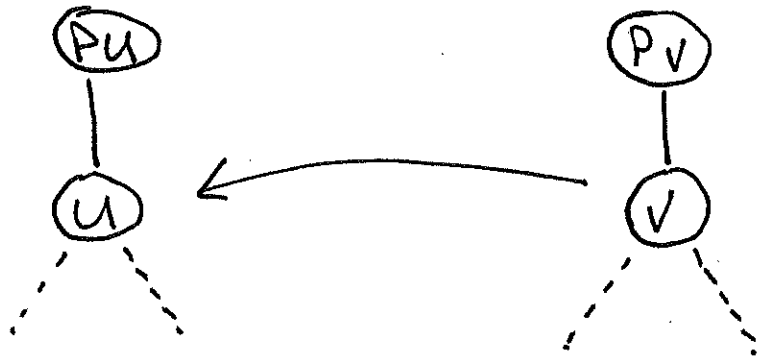


delete: 4 5



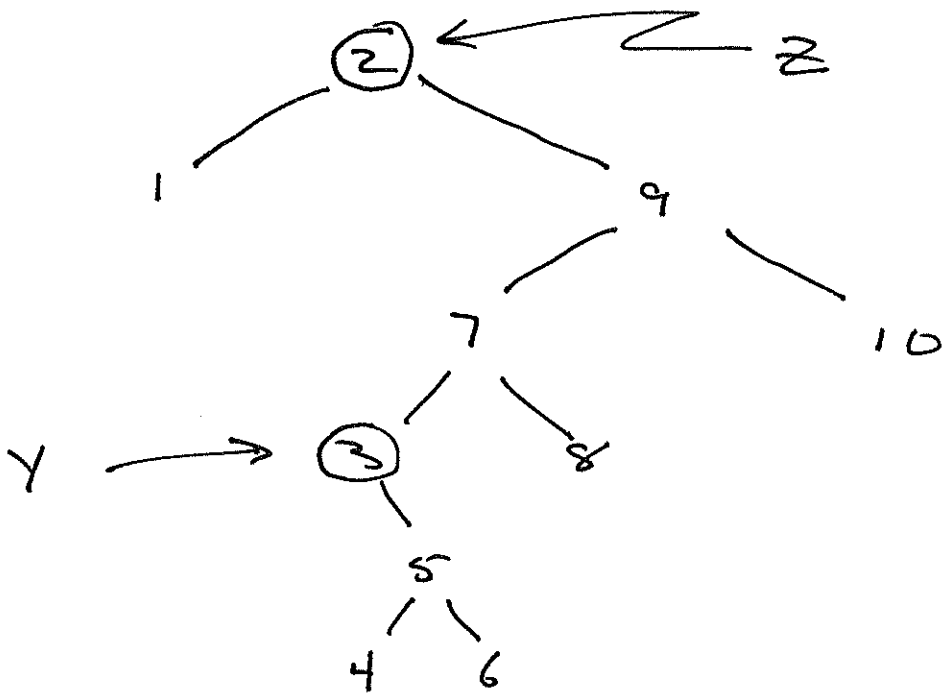
Pseudo-code for delete

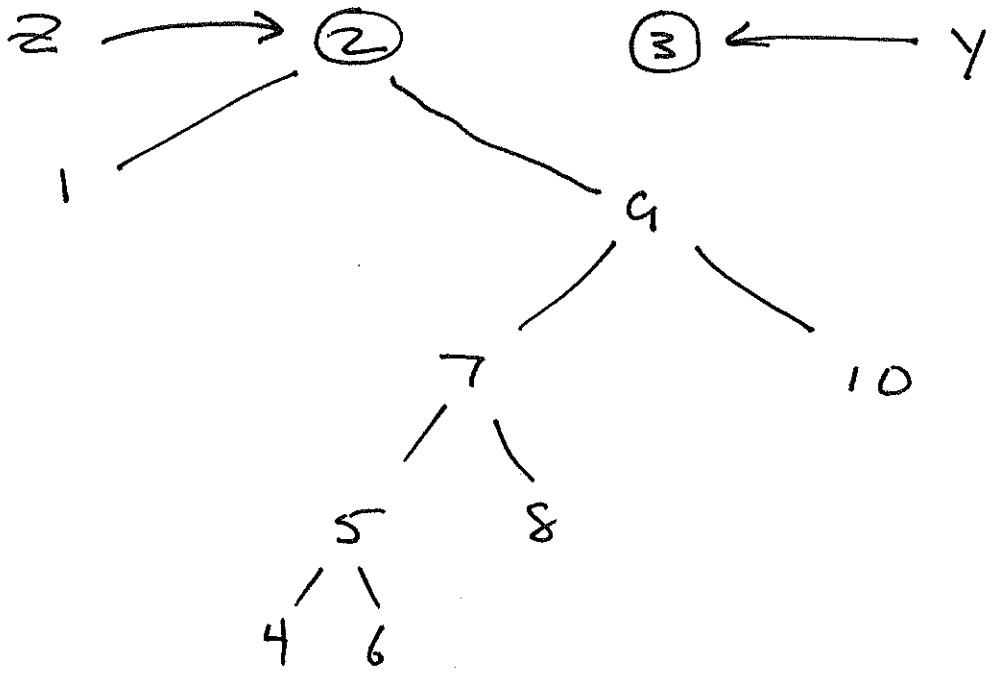
helper fcn. Transplant(T, u, v)

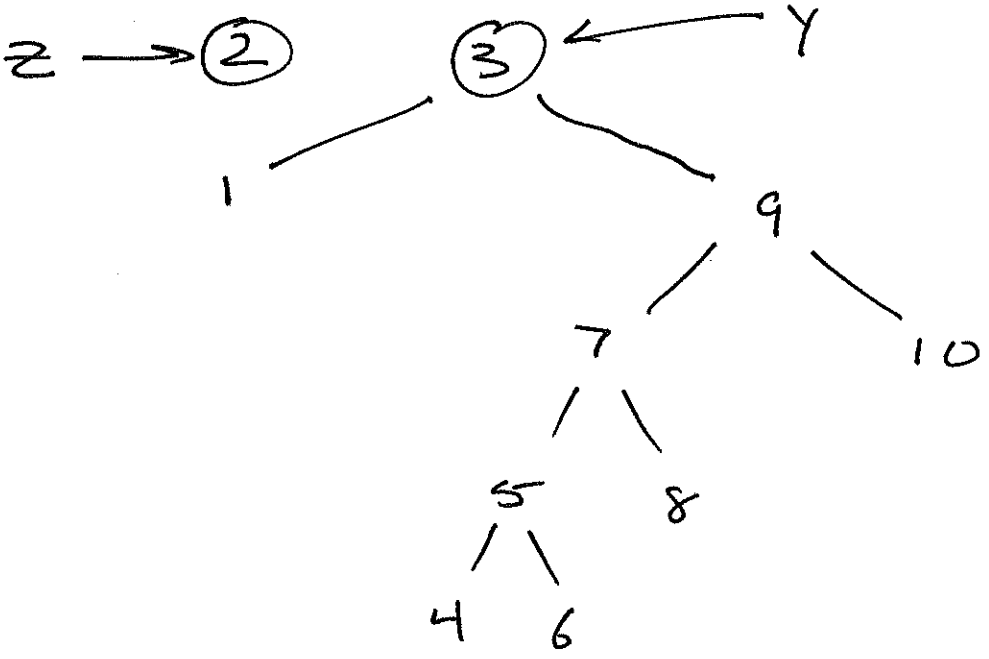


Delete (T, z) in case 3:

Ex







runtime = Θ(height(T))

Part:

forward & backward iterators

analogy: List ⟷ Dictionary

List	Dictionary
moveFront()	begin()
moveBack()	end()
get()	currentKey() / currentVal()
moveNext()	next()
movePrev()	prev()