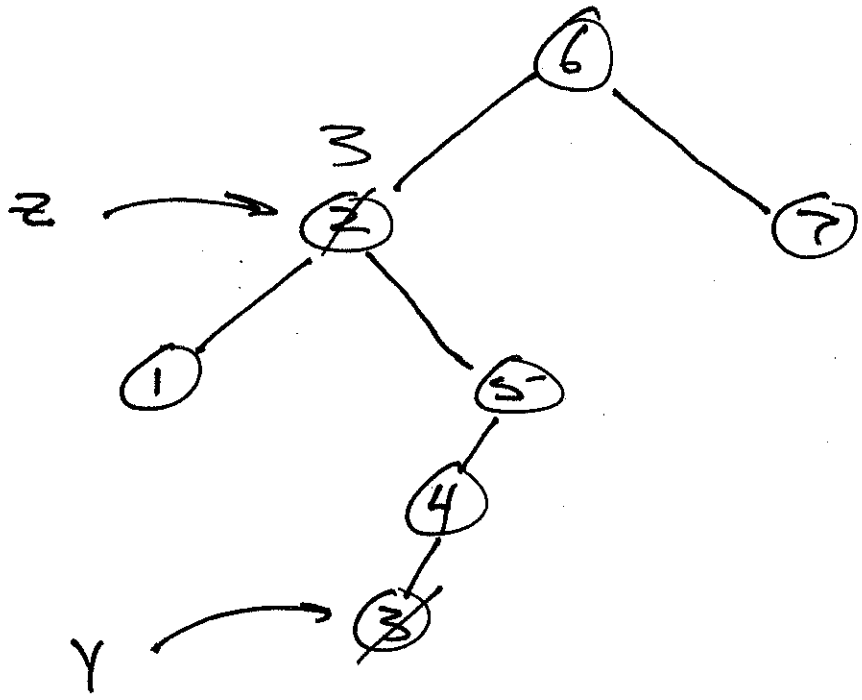
CSA 101 2-24-22

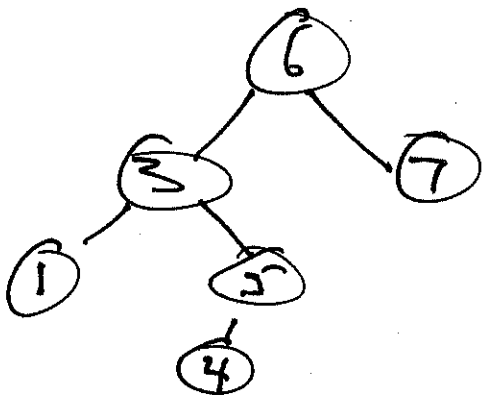
Pa7 : ext 1 day

Delete : case 3 : z has 2 children



successor of z

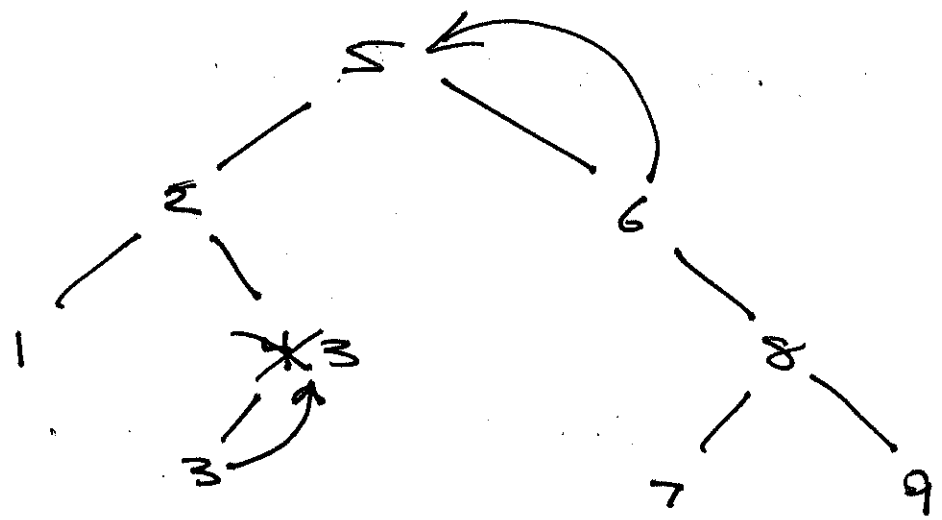
result of delete(z)

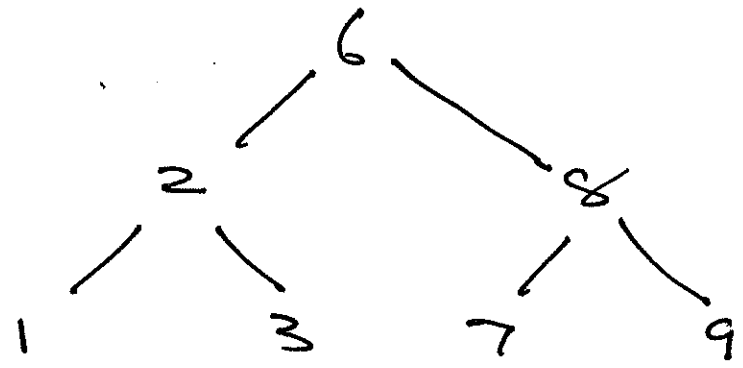


Ex. Insert into an empty tree

Keys: 5, 2, 6, 4, 3, 8, 7, 1, 9

then delete keys: 4, 5





Part

analogy with list ADT : forward and backward iteration

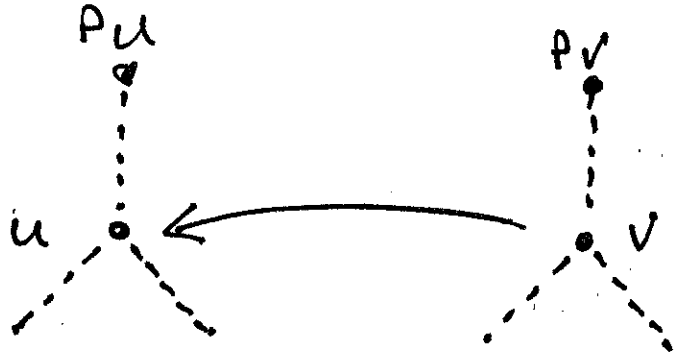
list	Dictionary
moveFront()	begin()
moveBack()	end()
get()	currentKey() / currentVal()
moveNext()	next()
movePrev()	prev()

To iterate over a Dictionary (forward)

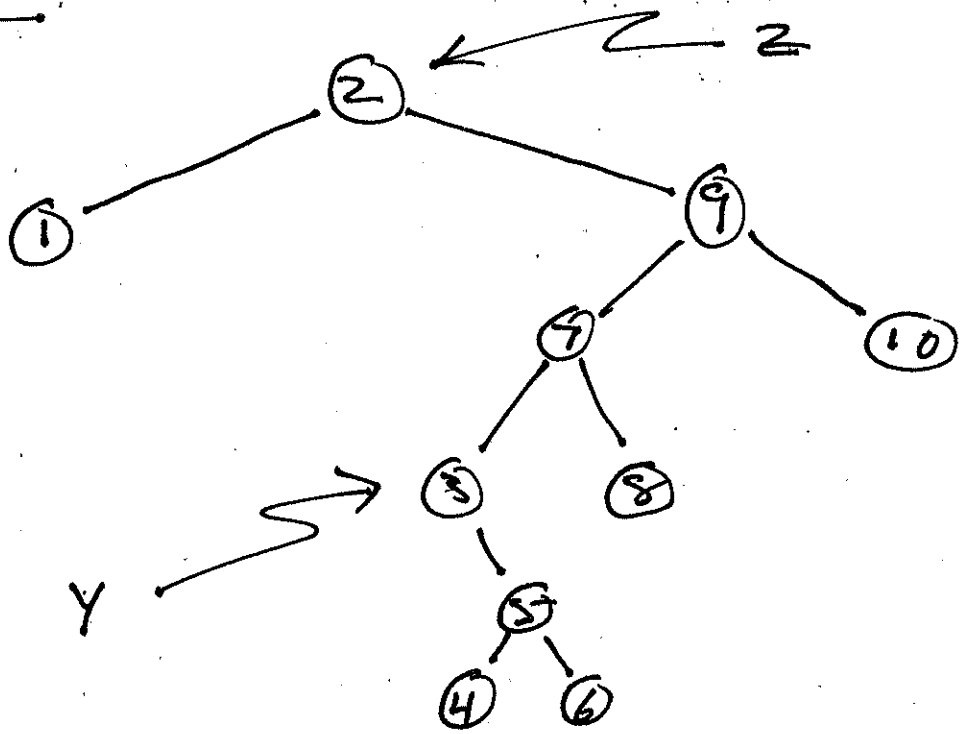
```
D.begin()
while(D.hasCurrent()){
   // use currentKey() or currentVal()
   next();
}
```

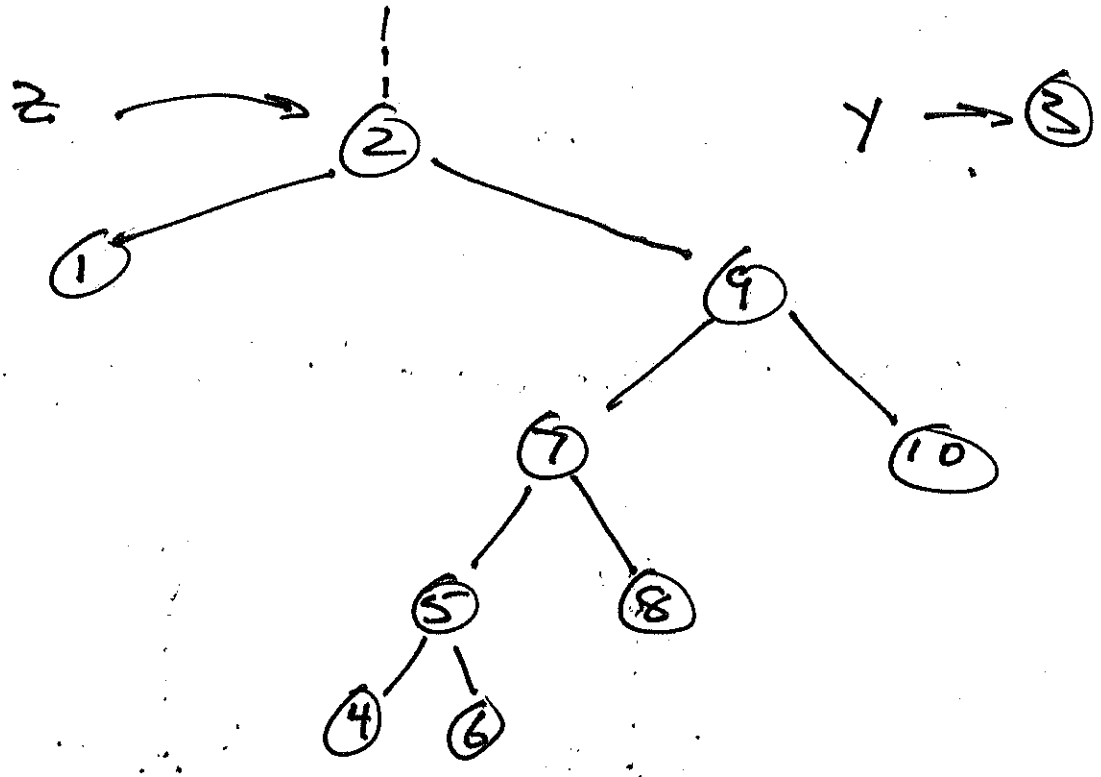
Pseudo-code for Delete(T)

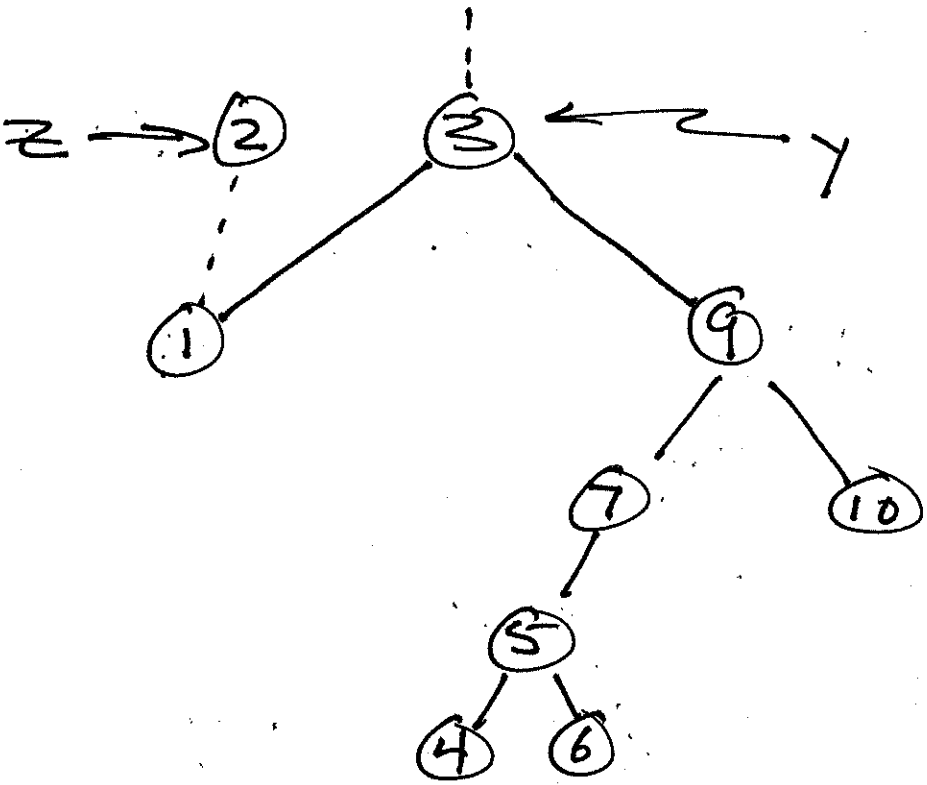
helper: Transplant(T, u, v)



Case 3:




z
2
y
3
1
9
7
10
5
8
4
6


z
2
3
y
1
9
7
10
5
4
6

what if you must insert keys in some specific order?

Red-Black Trees (RBT)

(chapter 13)

A RBT is a BST that also satisfies the RBT properties.

Convention

The leaves in a RBT are considered to be the nil children of key-bearing nodes.

RBT Properties:

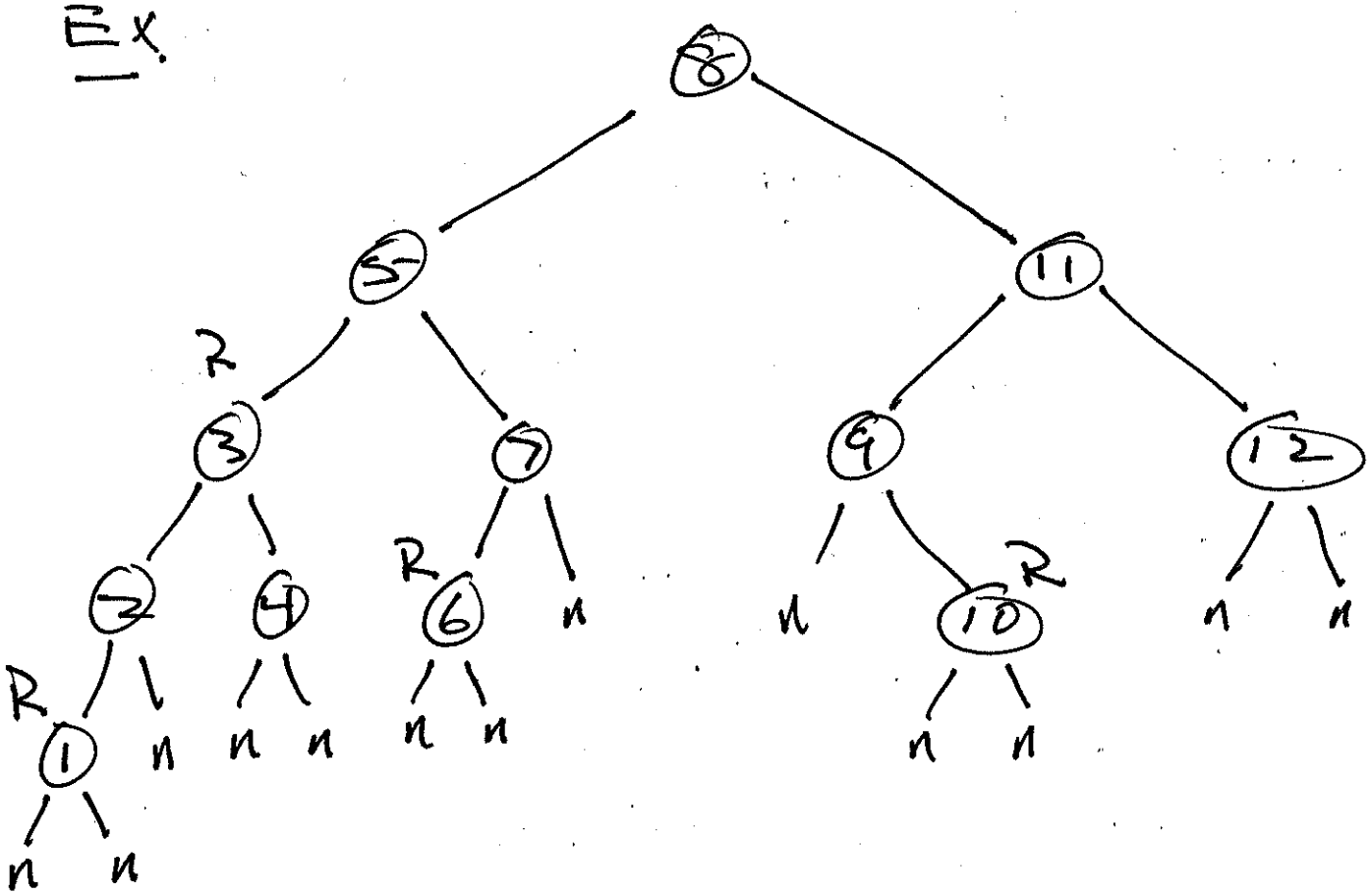
1. each node has a color: red or black.
2. root is black
3. each leaf (i.e. nil child) is black.
4. every red node has 2 black children (one or both could be nil.)
5. to any node x, every descending path from x to a leaf has the same # of black nodes.

Defn

The black height of a node x, denoted $bh(x)$, is the # of black nodes in any desc. path to a leaf (not counting x),

note: this is well defined By RBT Prop. (5).

Ex.

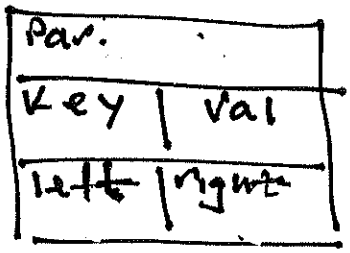


nodes	black height
8	3
5, 11, 3	2
1, 2, 4, 6, 7, 9, 10, 12	1
nil leaves	0

Par: stub

what does nil look like?

nil:



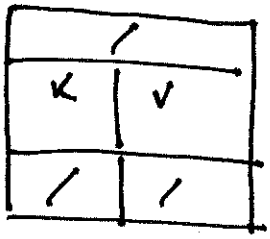
suggestion:

set key = "\000"

↖ null character. could use any comb. of non-printable chars.

set val = ??

what about Parent, left, right ?



or