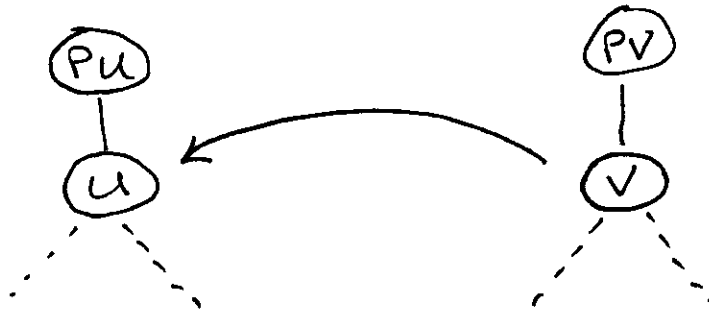


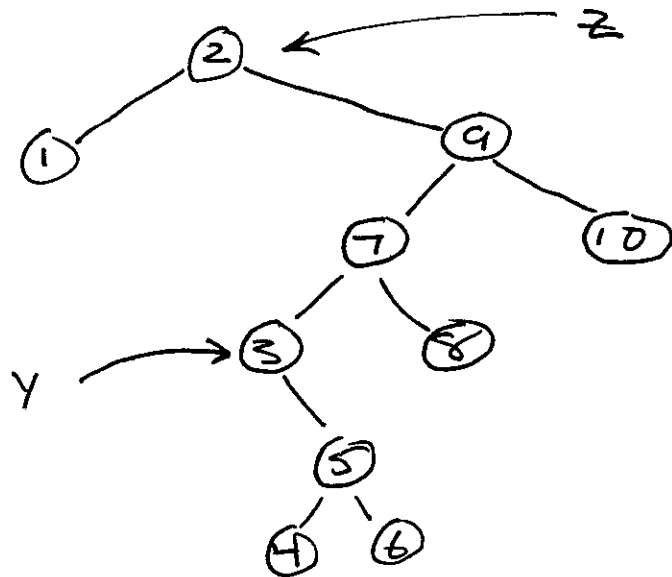
PaS: ext. 3 days.

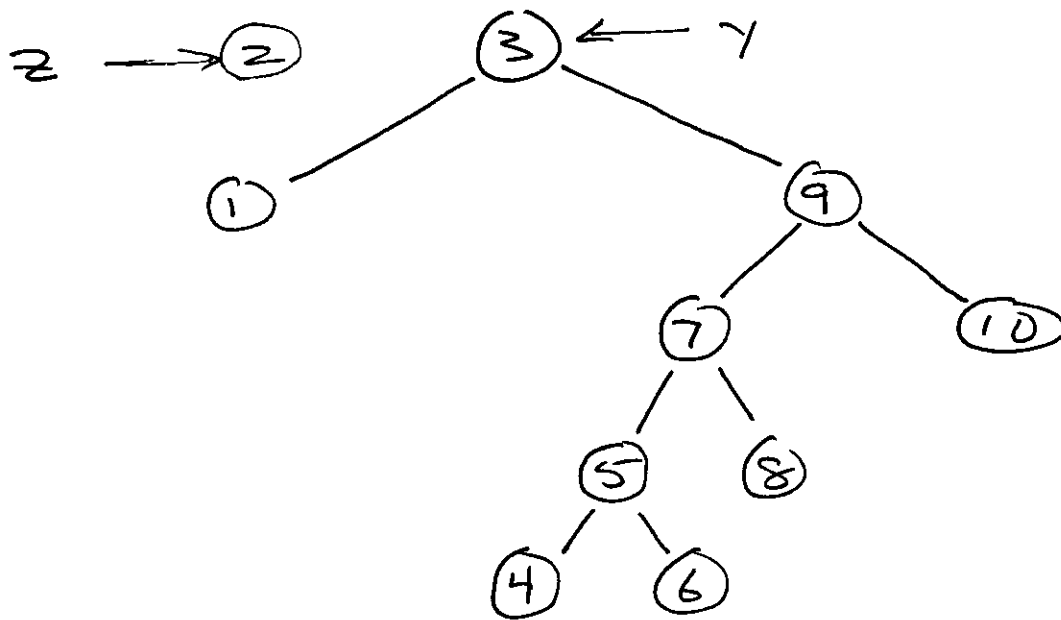
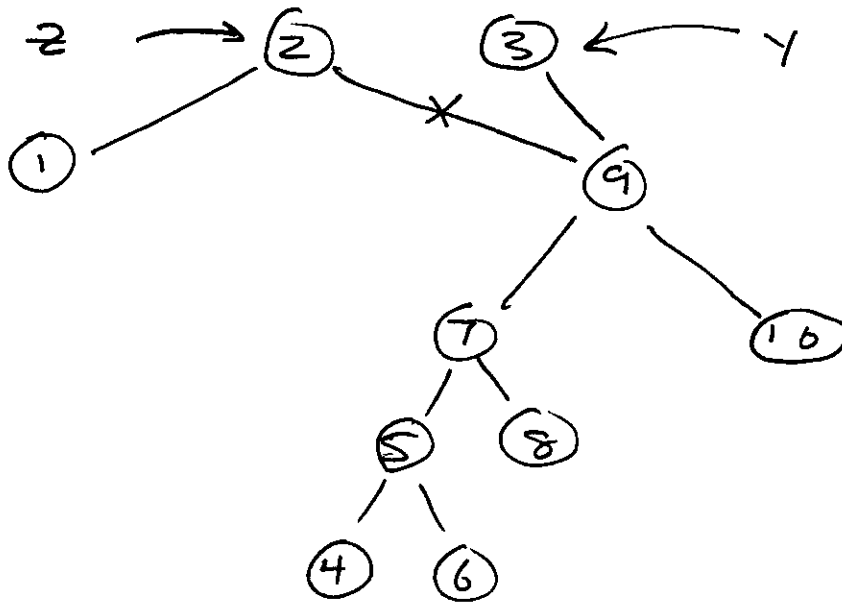
Delete in a BST:

Transplant(T, u, v)



Delete(T, z)
Ex(Case 3)





Pa 6:

- forward $\hat{=}$ backward iteration
analogy

list

moveFront()

moveBack()

get()

moveNext()

movePrev()

Dictionary

beginForward()

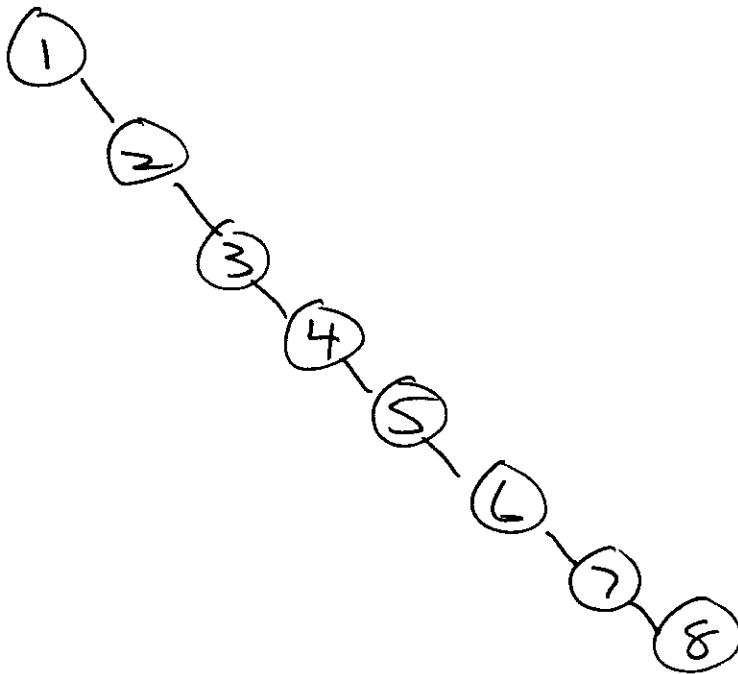
beginReverse()

{ currentKey()
currentVal()

next()

prev()

Problem: If we insert keys
1 2 3 4 5 6 7 8, in order,
we get



Runtime of insert(), lookup(), delete()
are all (worst case)

$$O(\text{height}(T))$$

Solution: RedBlack Tree (RBT)

LS

Goal: Balance a BST so that insert, lookup, delete are efficient

Chap. 13 (RBT)

Defn A RBT is a BST that satisfies the RBT Properties below (in addition to BST properties.)

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

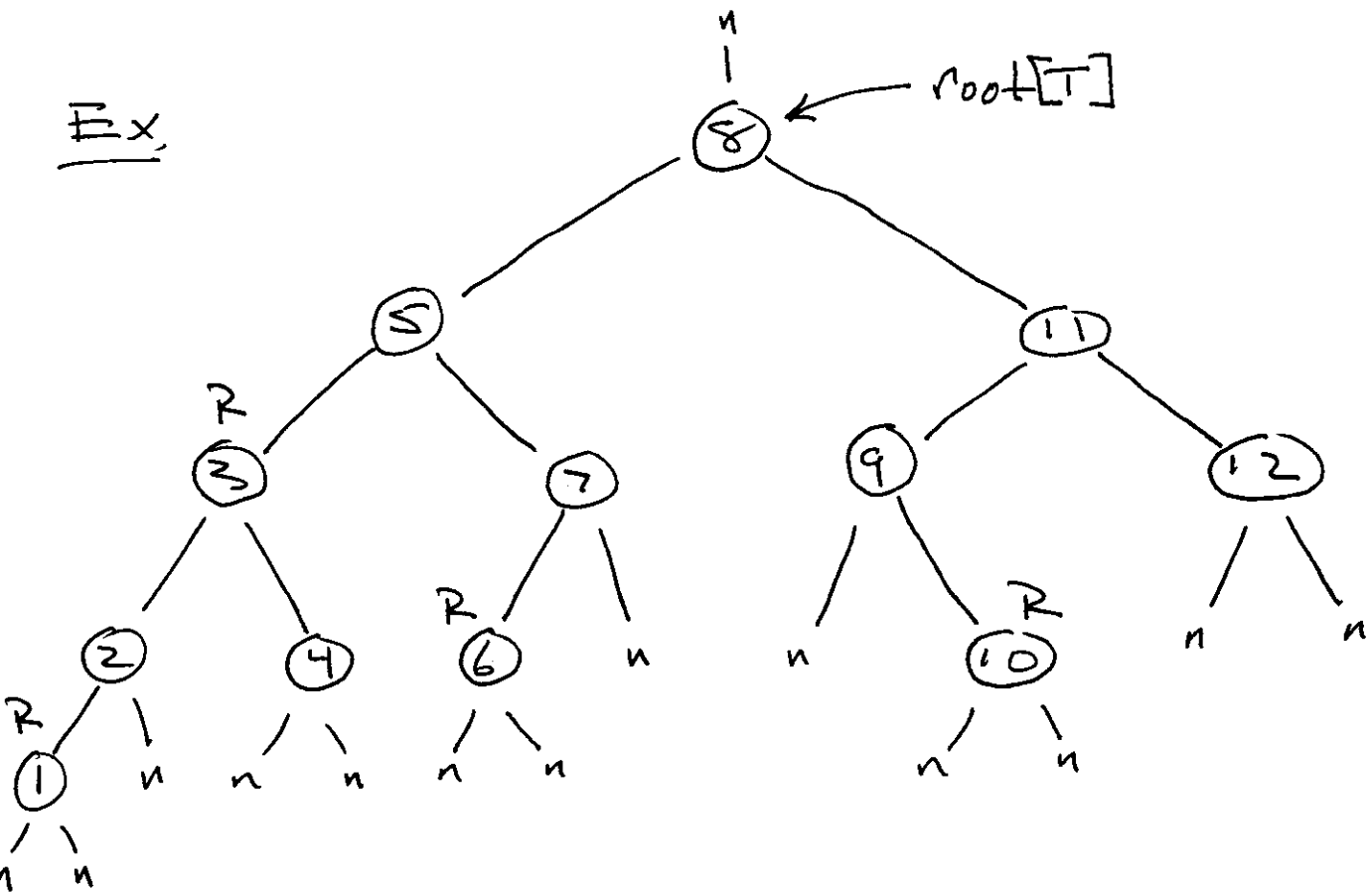
RBT Properties

1. each node has a color attribute, Red or Black.
2. $\text{root}[T]$ is Black
3. each leaf (nil child) is Black.
4. Every Red node has 2 Black children (one or both of which may be nil.) Equivalently, Red nodes do not have Red children.
5. for any node x in T , every descending path from x to a leaf contains the same number of black nodes.

Defn The Black height of x , denoted $bh(x)$, is the # of black nodes in any descending path from x to a leaf (not counting x itself.)

Ex,

7



node

black height

8

3

5, 11, 3

2

1, 2, 4, 6, 7, 9, 10, 12

1

all nil leaves

0

Observe

$bh(x) = 0$ iff $height(x) = 0$ iff x is leaf

↑
absolute height

Theorem

A $\mathcal{RRT}$ with n internal (non-null) nodes and height h satisfies

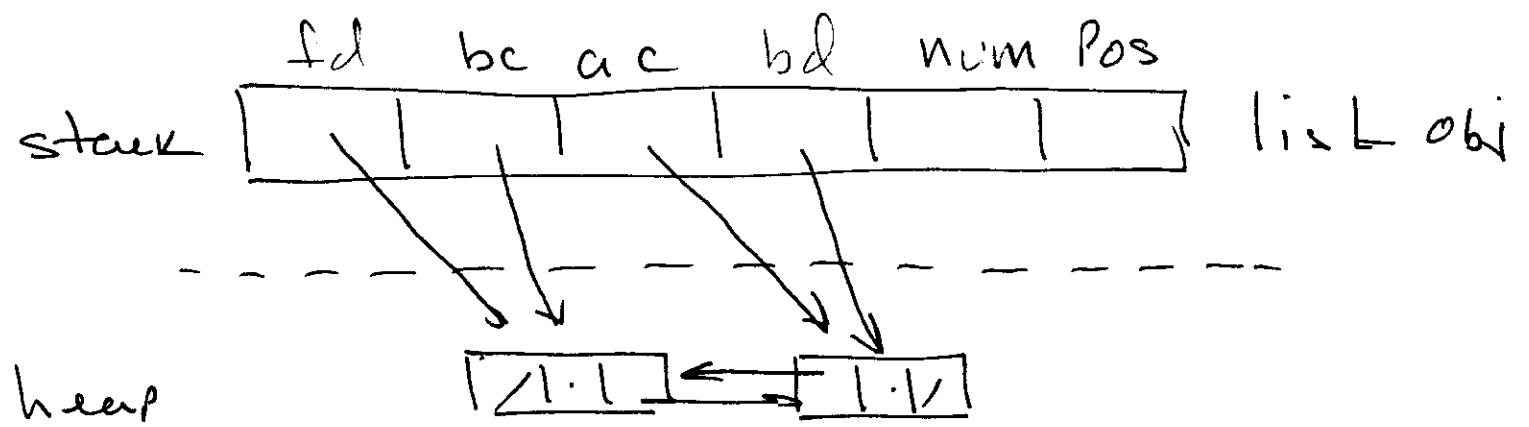
$$h \leq 2 \lg(n+1)$$

Ex (Previous)

$$h=5, n=12$$

$$2 \cdot \lg(n+1) = 2 \cdot \lg(13) > 6 > 5 = h \quad \checkmark$$

Pass 1: stuff



in client List L;

```

int main() {
    List L;
}

```

client view:

2 5 | 3 7

insertBefore (6)

2 5 6 | 3 7