CSE 101 11-25-25

PA7: ext. 2 days (last)

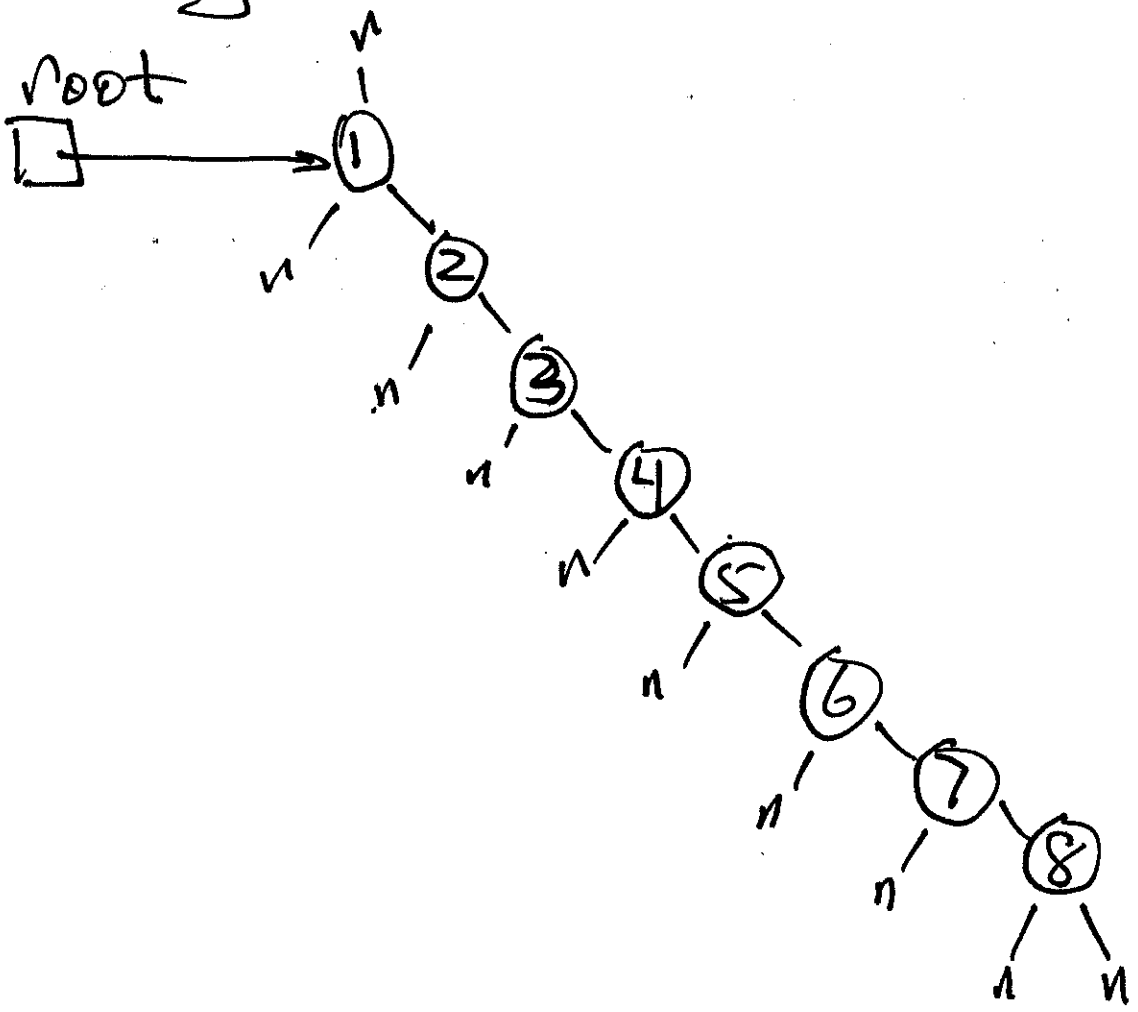
PA8: due next Tue.

SETS: Due Sun 12/7 11:59 PM

Problem if we insert keys

1, 2, 3, 4, 5, 6, 7, 8 (in order)

We get

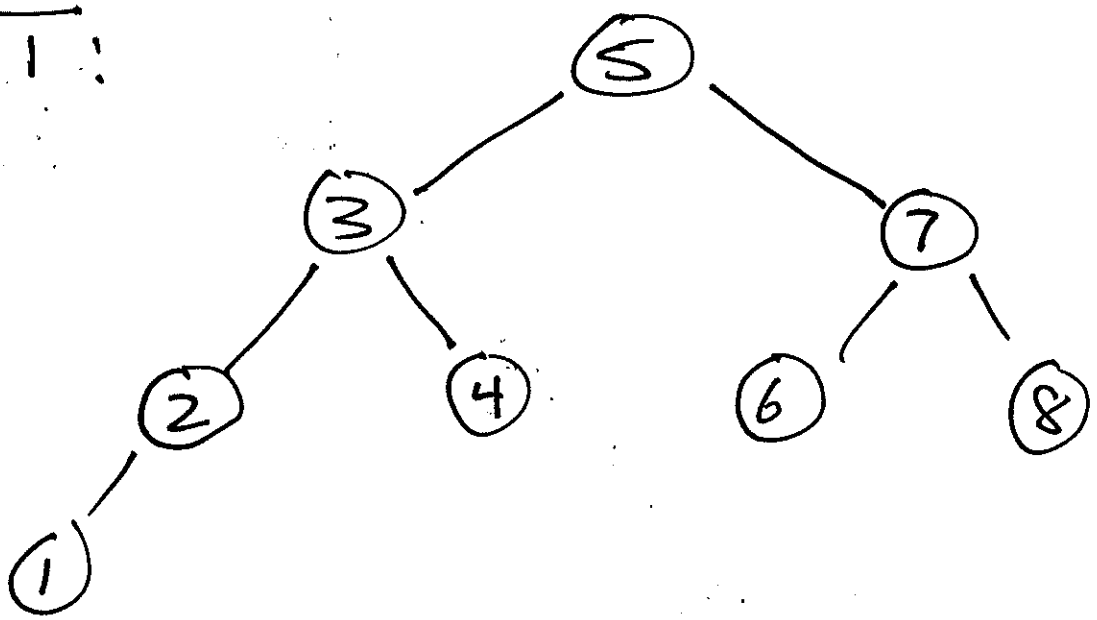


Runtime: $\Theta(n)$ where n = # keys.

Diff. order:

5, 3, 7, 4, 2, 6, 8, 1

T:



Worst case runtime of search: $\Theta(\text{height}(T))$

4 in this case

Fact: If a BST contains n nodes then

$$\lfloor \lg(n) \rfloor \leq \text{height}(T) \leq n - 1$$

↑ requires Proof

>0

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

↑ Balanced Tree

Chapter 13: Red Black Trees (RBT)

Goal: Balance BST so that runtimes are $\Theta(\log n)$

Def: a RBT is a BST that satisfies RBT properties.

Convention
leaves in an RBT are considered to be nil children of key-bearing nodes.

RBT Properties

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

Def:

The Black height of x

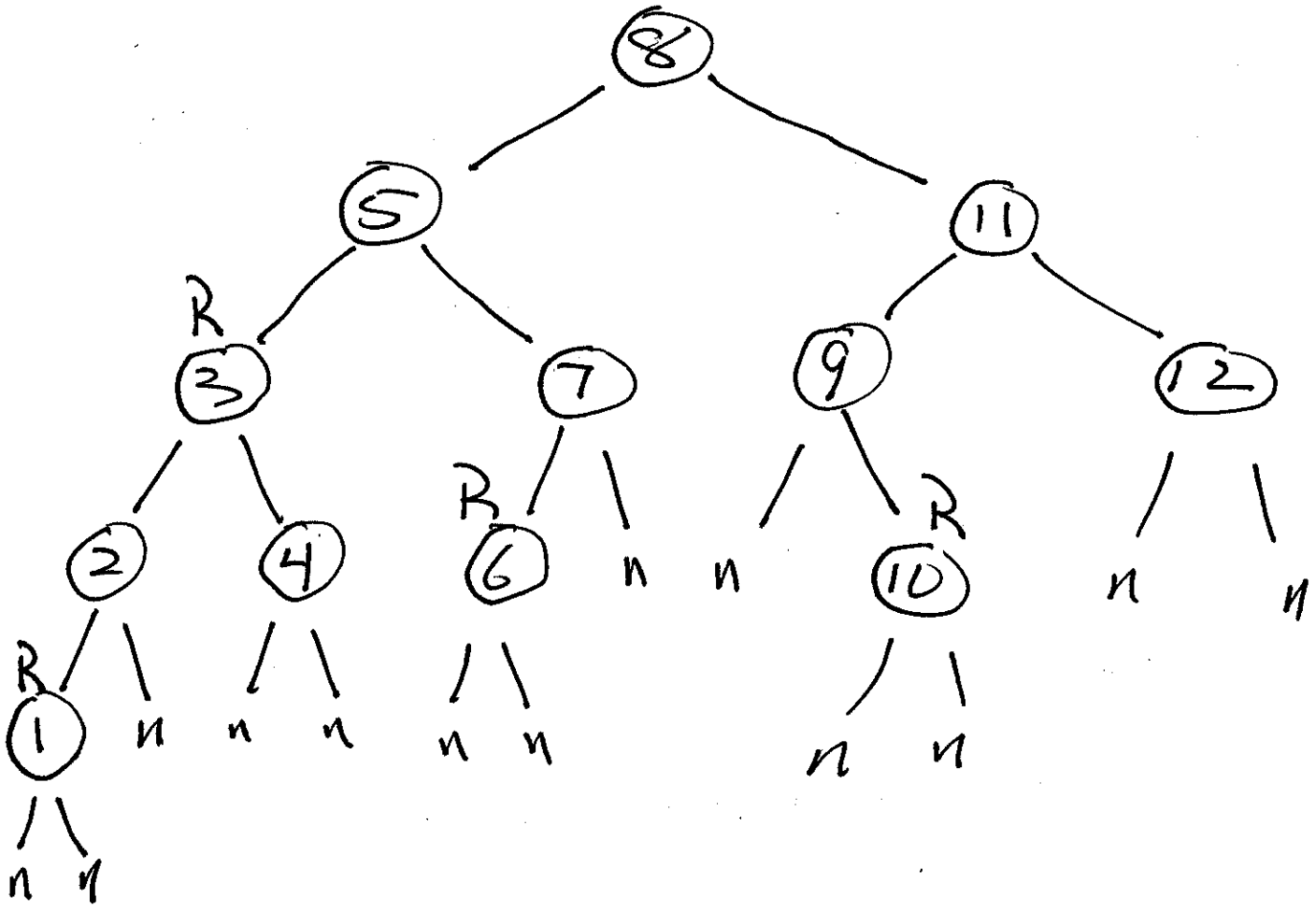
bh(x)

is the # of black nodes from x to a descendant leaf (not counting x).

Note: bh(x) is well defined by axiom (5).

note height(x) is called absolute height now.

Ex.



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

note

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

iff x is a leaf

Exercise

find all assignments of colors in last example that satisfy the RBT Properties

Theorem

An RBT with n internal nodes (i.e. non-nil nodes) and height h satisfies

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

Rmks

- any BST satisfies $h \geq \lfloor \lg n \rfloor$, so a RBT satisfies

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

$$\therefore \text{height}(T) = \Theta(\log n)$$

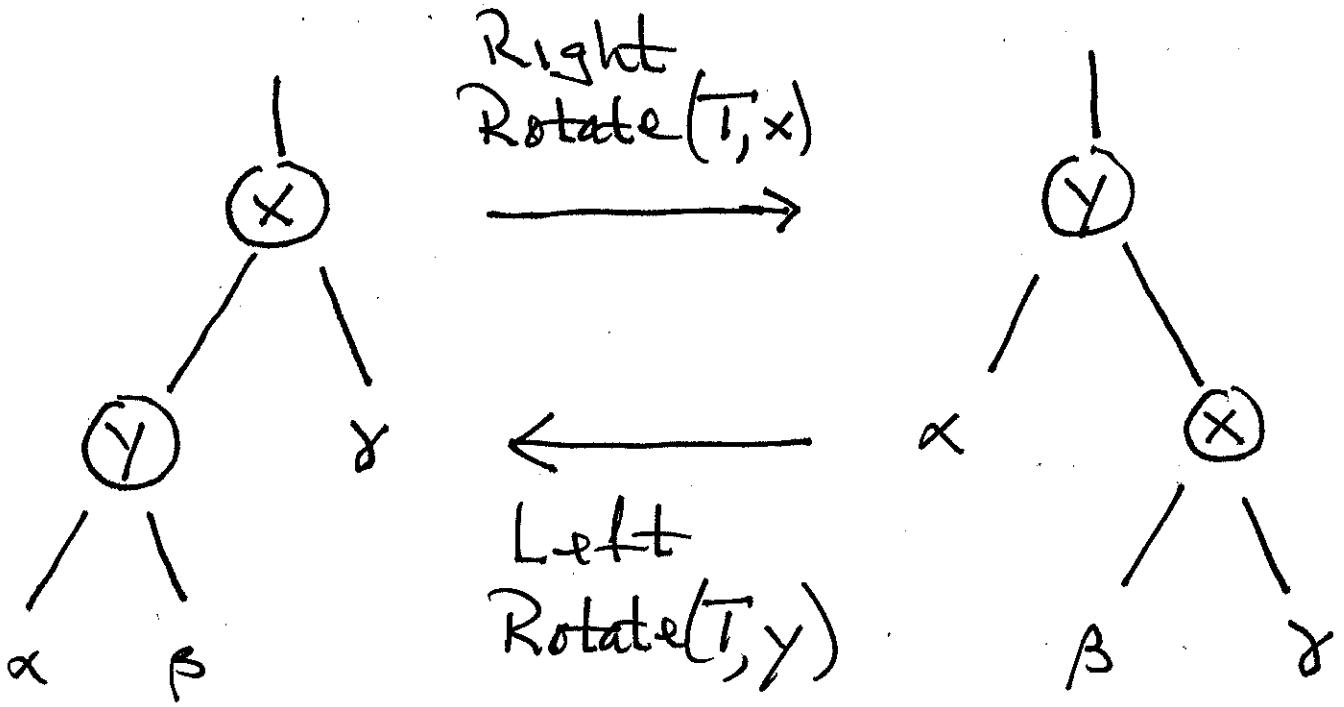
i.e. T is Balanced.

- Insert() and Delete() in a BST do not Preserve RBT Properties

- Insert() & Delete() can be altered to efficiently Preserve RBT Properties.

13.2 Rotations

Picture:



note both rotations preserve the BST properties

$$\alpha \leq y \leq \beta \leq x \leq \gamma$$

3.3 Insertion

- Start by doing a BST insert, and color the new node Red.
- Goal: fix RBT Property (4) if it is violated.
- See RBT-insert-fixup.