CSE 101 5-26-26

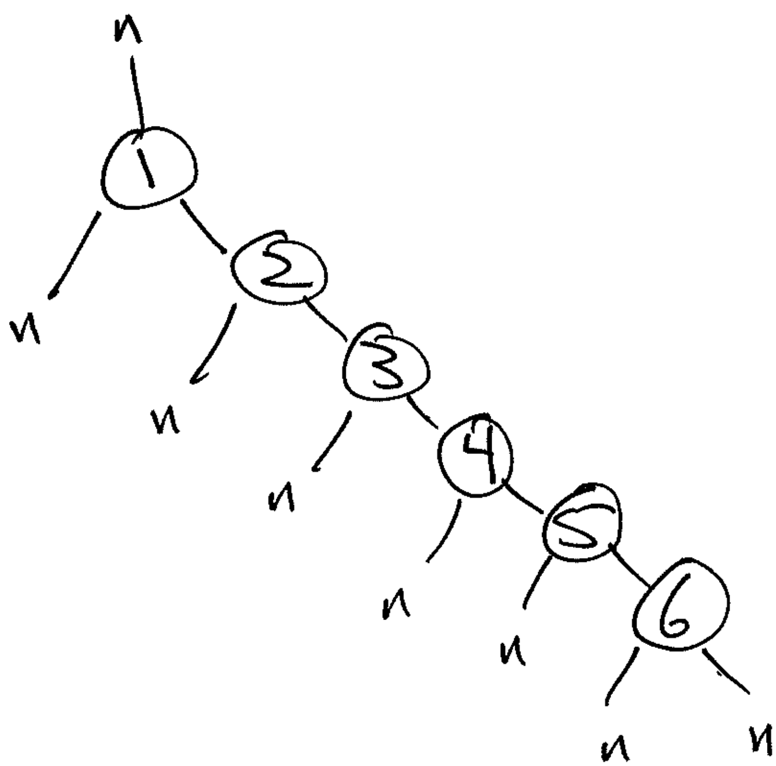
Pa7: ext. 1 more day Fri.

Pa8 : Posted

Chap. 13: Red-Black Trees (RBT)

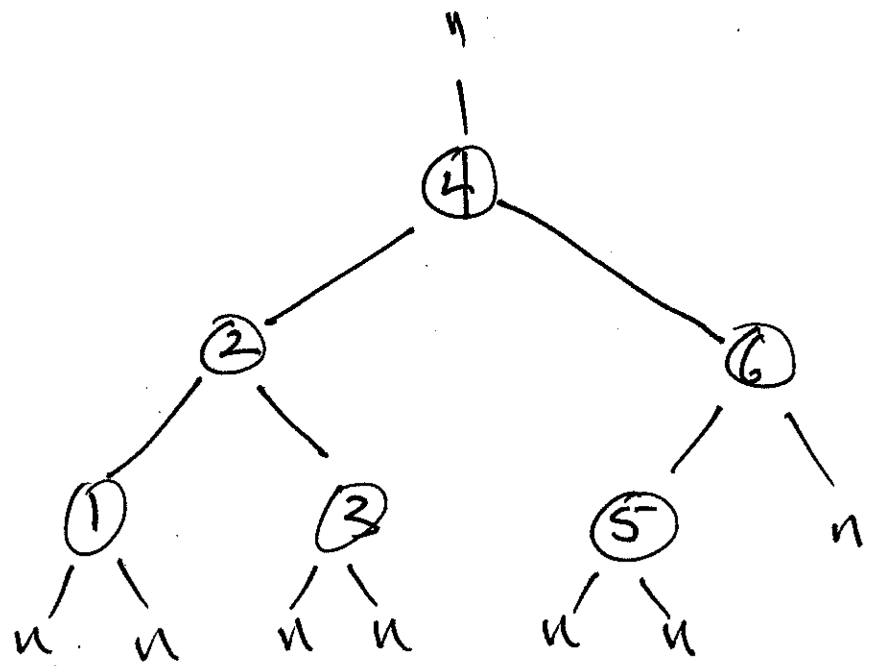
Recall Problem with BSTs

Bad search structure



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

Good search structure



'Balanced' means

$$height(T) = \Theta(\log n)$$

RBT Properties

1. Each node is Red or Black.
2. root is Black.
3. Each leaf is Black

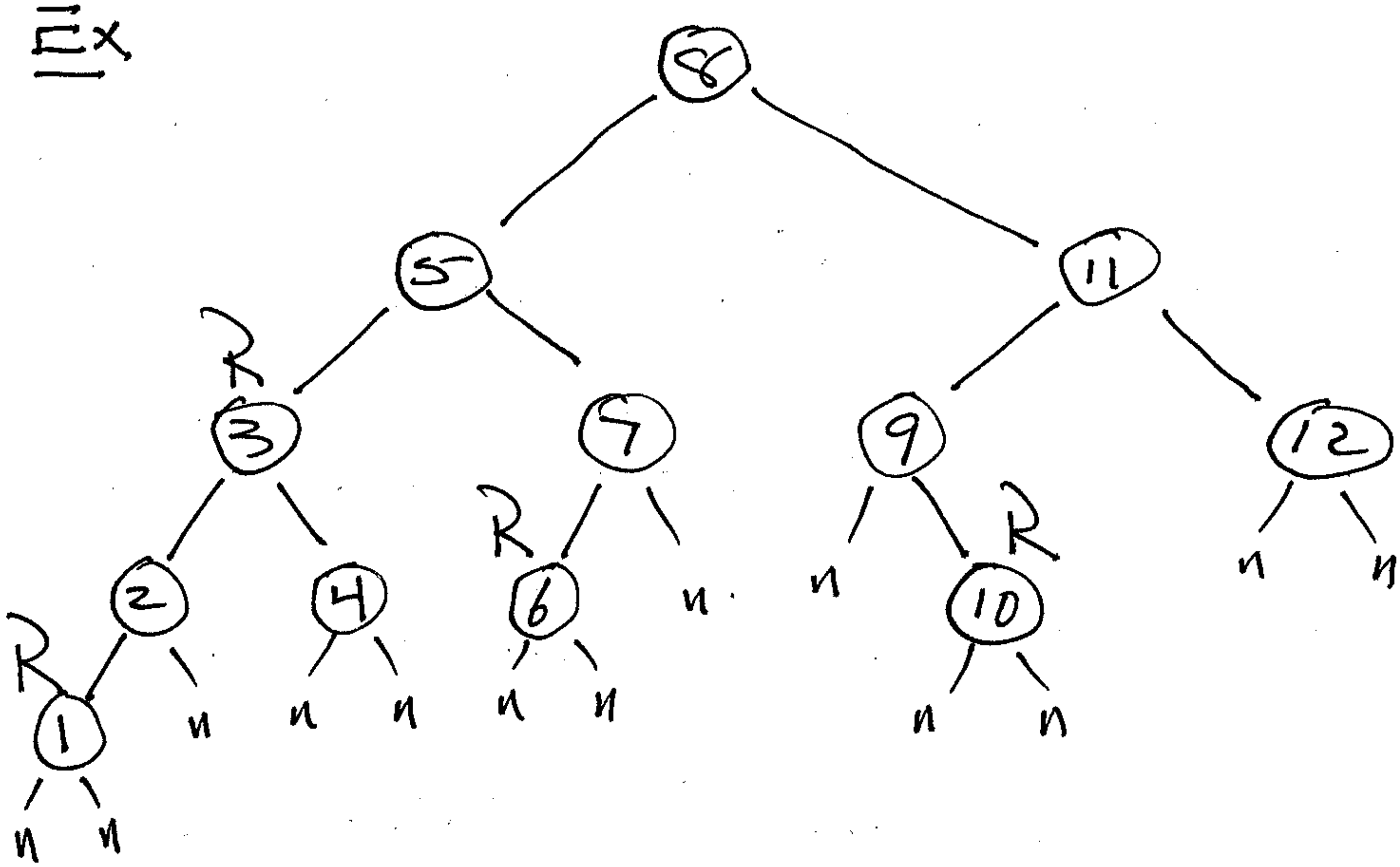
   Convention: leaves are nil children. internal nodes are key bearing nodes.
4. Every Red Node has 2 Black children (one or both of which may be nil)

5. For any node x, every descending path to a leaf contains the same # of Black nodes.

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

<u>Recall</u>: height(x) is the maximum length of any descending path from x.

<u>Ex</u>



Node	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

Theorem

A RBT with n internal (i.e. key bearing) nodes and height h satisfies

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

Remark

Any binary tree satisfies

$$h \geq \lfloor \lg n \rfloor$$ 'Proof' later

Thus in a RBT:

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

Thus

$$\Omega(\log n) \le \text{height}(T) \le O(\log n)$$

RBT

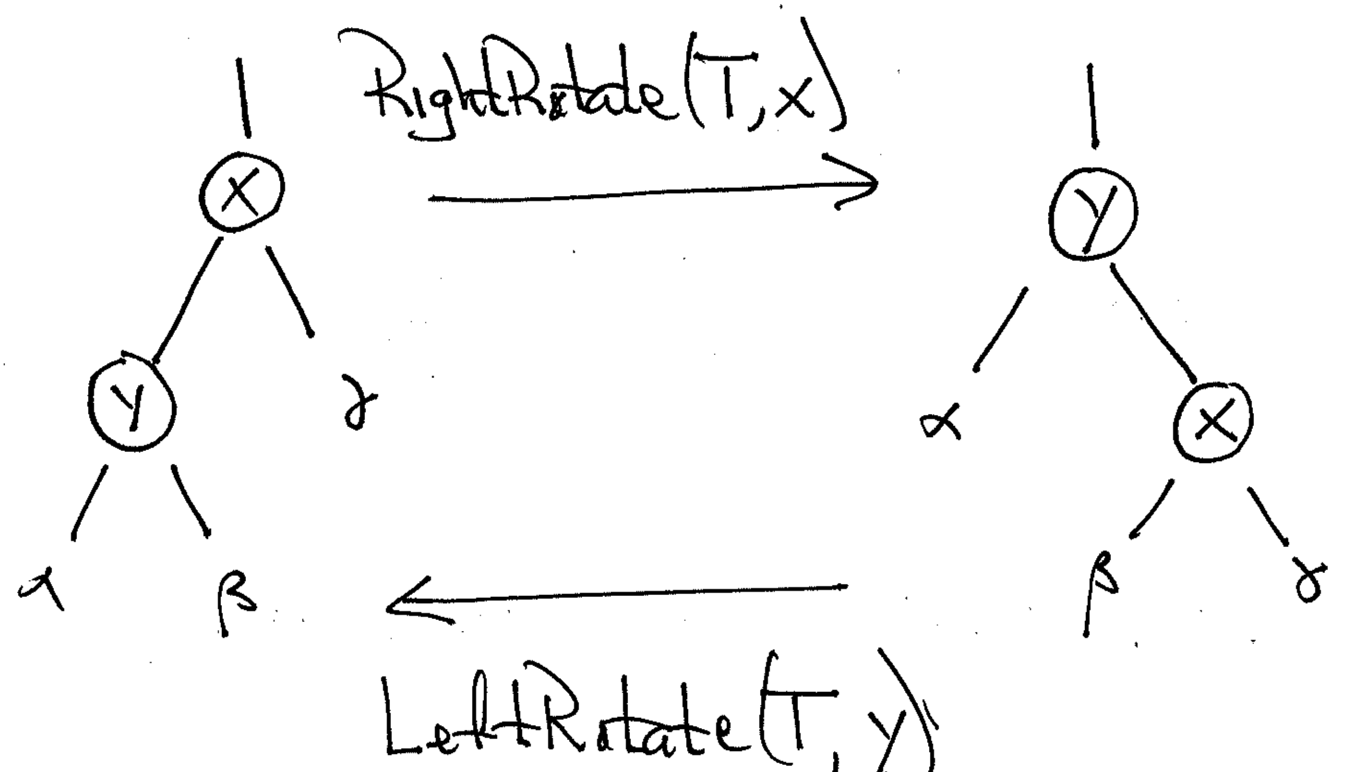
$$\therefore \quad \text{height}(T) = \Theta(\log n)$$

$\therefore$ Runtime of Queries is

$$\Theta(\log n).$$

13.2 Rotations

Picture



Summary of RightRotate(T, x)

$$\beta \left\{ \begin{array}{l} \bullet \ \ x.\text{left} = y.\text{right} \\ \bullet \ \ y.\text{right}.\text{Parent} = x \end{array} \right.$$

$$\text{Parent} \left\{ \begin{array}{l} \bullet \ \ y.\text{Parent} = x.\text{Parent} \\ \bullet \ \ x.\text{Parent's (left or right) child} = y \end{array} \right.$$

$$x, y \left\{ \begin{array}{l} \bullet \ \ y.\text{right} = x \\ \bullet \ \ x.\text{Parent} = y \end{array} \right.$$

13.3 Insertion

next time . . .