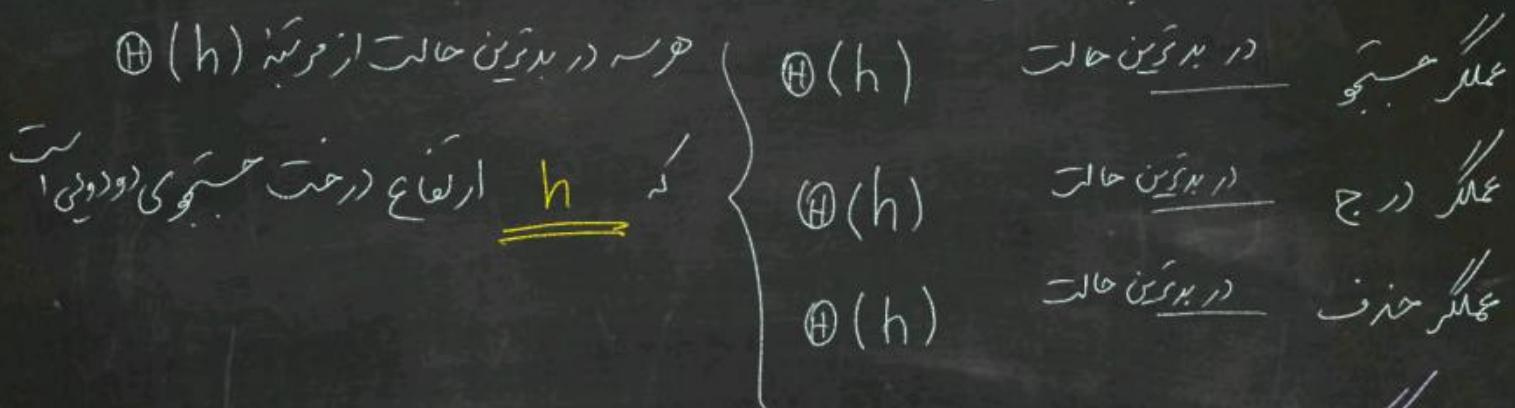


Balanced Binary Search Tree درخت جستجوی دودویی مُوازن

در سایر درخت جستجوی دودویی عکس‌های اصلی ساختار را داده‌ایم پوچیده، مینی:

$\Theta(h)$ عمل جستجو در برابرین حالت از مرتبه



* می‌دانیم کسره تغیرات ارتفاع درخت جستجوی دودویی می‌تواند بین

$$\lg n \leq h < n$$

اگر $h = O(n)$ باشد معنای آن اینست که ساختار داده‌ای درخت جستجوی دودویی است.

درخت جستجوی دو دلیل متوزن

در سبد درخت جستجوی دو دلیل عکس های اصلی ساختار را (دایرکت) نمایم، صنف:

(۱) عکس جستجو $\Theta(h)$ در برخین حالت

(۲) عکس درج $\Theta(h)$ در برخین حالت

(۳) عکس حذف $\Theta(h)$ در برخین حالت

* بنابراین نمایش نه ساختار را دایرکت جستجوی بهتر باشد

خواهد نصیحت (استفاده) $h = O(\lg n)$

(درخت جستجوی دو دلیل که نصیحت است)، آن برآوردها درخت متوزن تایید می شود

→ درخت کامل / درخت هموزن اس

$$n = 2^{h+1} - 1 \Rightarrow h = O(\lg n)$$

→ اما لزوماً سطح درخت هموزن، درخت کامل نیست

می‌توان برواری $h = O(\lg n)$ را به سهوی (لکری نیز) تضمین کرد:

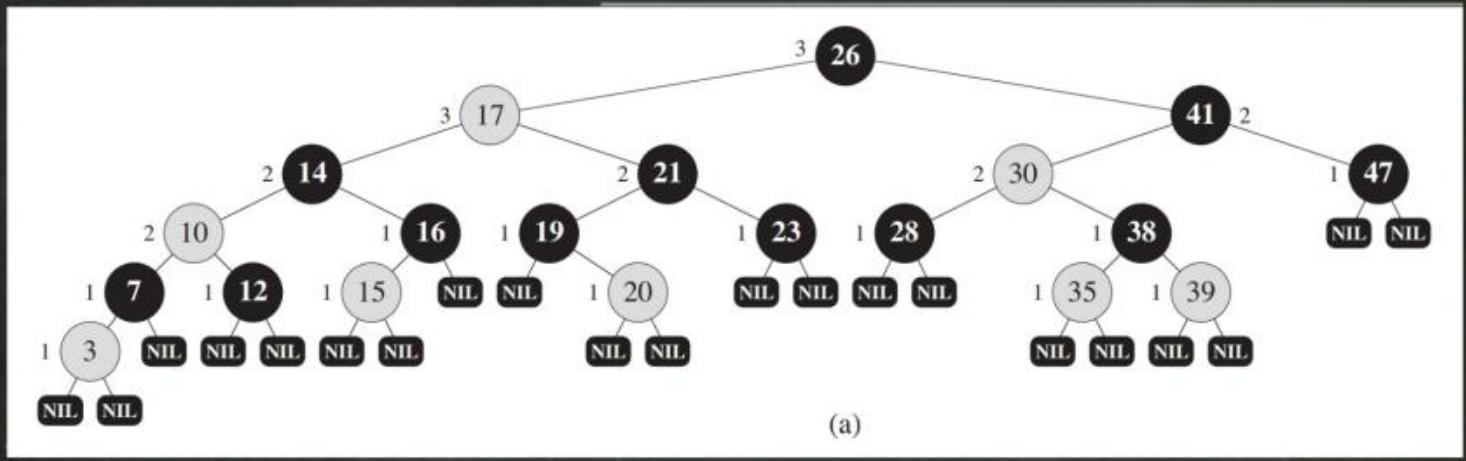
سلع درخت AVL. سه تر درخت همواره طوری تبلداری می‌سود که
بازای هرگره درخت، اختلاف ارتفاع زیردرخت را که گرفته از فرزند پسر و زن درخت
لئے گرفته از فرزند راست آن گره حداقل ۱ باشد.
 $n > \text{fib}_0(h) \Rightarrow h = O(\lg n)$

درختهای حسنه‌بودی (و وی) متوالی

لیکٹریسٹ مہاری (فینا) میں رحمت حسینی دو دوستی میں

Red-Black Tree

ساختار داده‌ای درخت فرمز-سیاه

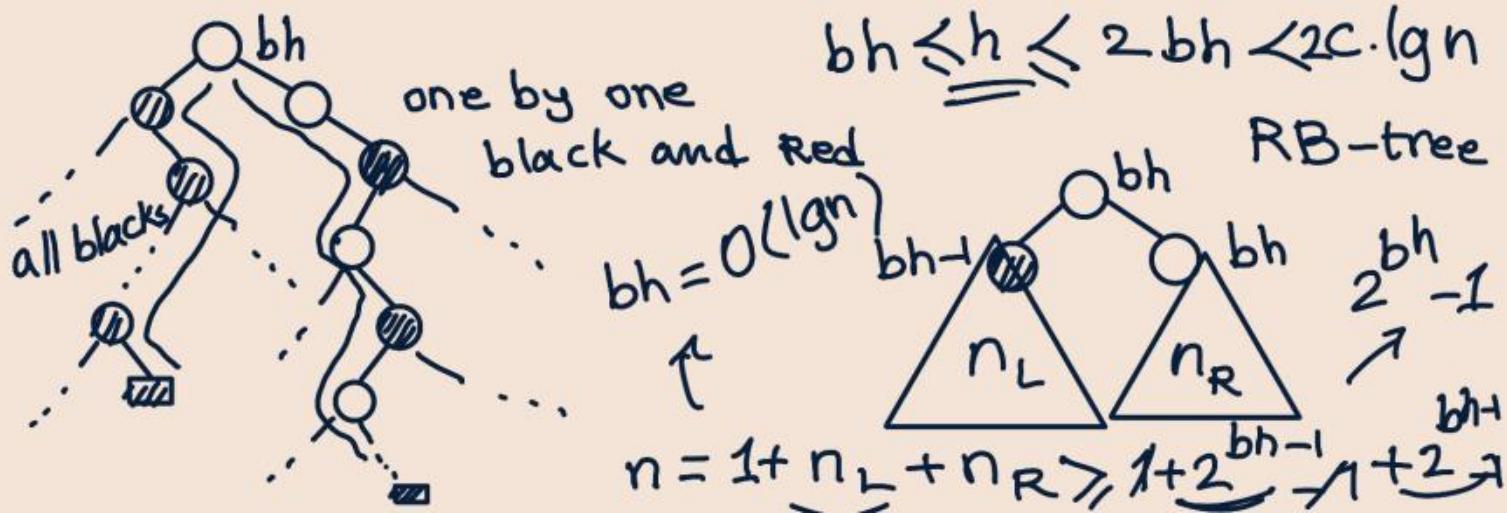


(a)

- ✓ 1. Every node is either red or black.
 - ✓ 2. The root is black.
 - ✓ 3. Every leaf (NIL) is black.
 - * 4. If a node is red, then both its children are black.
 - ** 5. For each node, all simple paths from the node to descendant leaves contain the same number of black nodes. → **black height, bh**
- $\Rightarrow h \leq 2 \lg n \Rightarrow h = O(\lg n)$

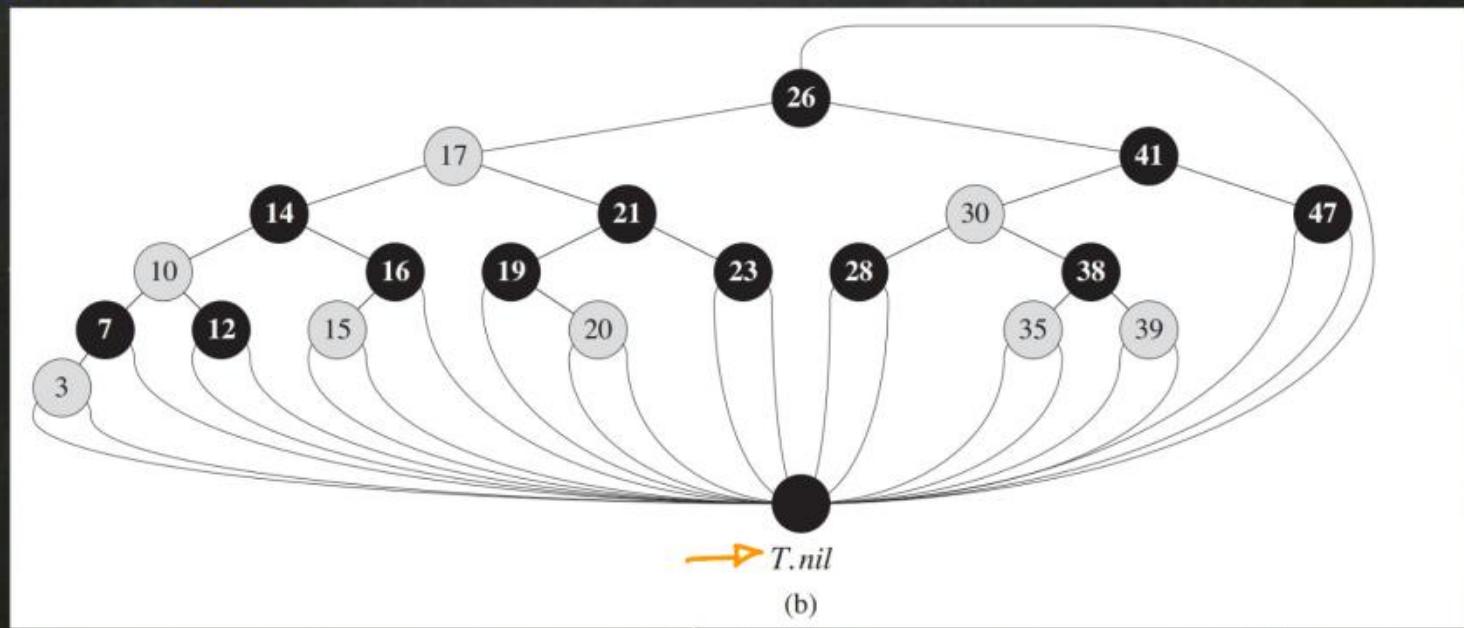
```
class RBnode extends BSTnode{
    color : (Red, Black)
};

class RBtree extends BST{
};
```



یک قرارداد برای پاده‌سازی ساختار داده‌ای درخت فرمز-سیاه

عنصر تابیان (گره تابیان) \leftarrow گره موجود
که همچنان خواهد داده شد



روابط هر سعی از نوع RBtree برای خود گره nil دارد.

class RBtree{

RBtree(){

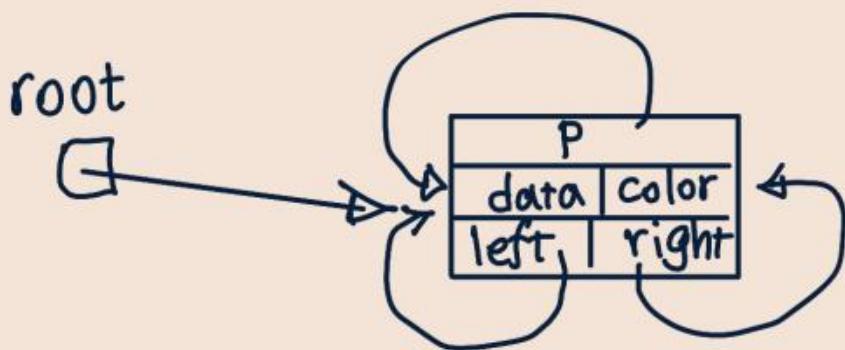
RBnode *nil = new RBnode;

nil->left = nil->right = nil->p = nil; nil->color =
BLACK;

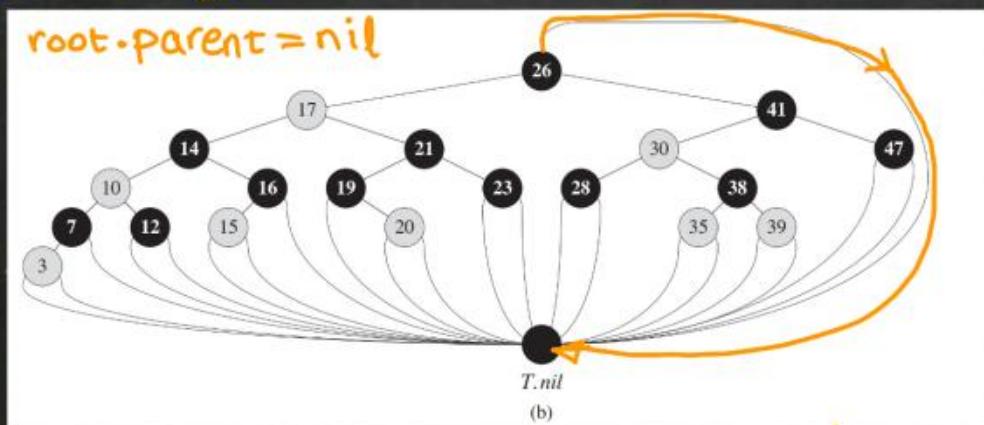
root = nil;

}

درست رفع سعی سازنده طاری



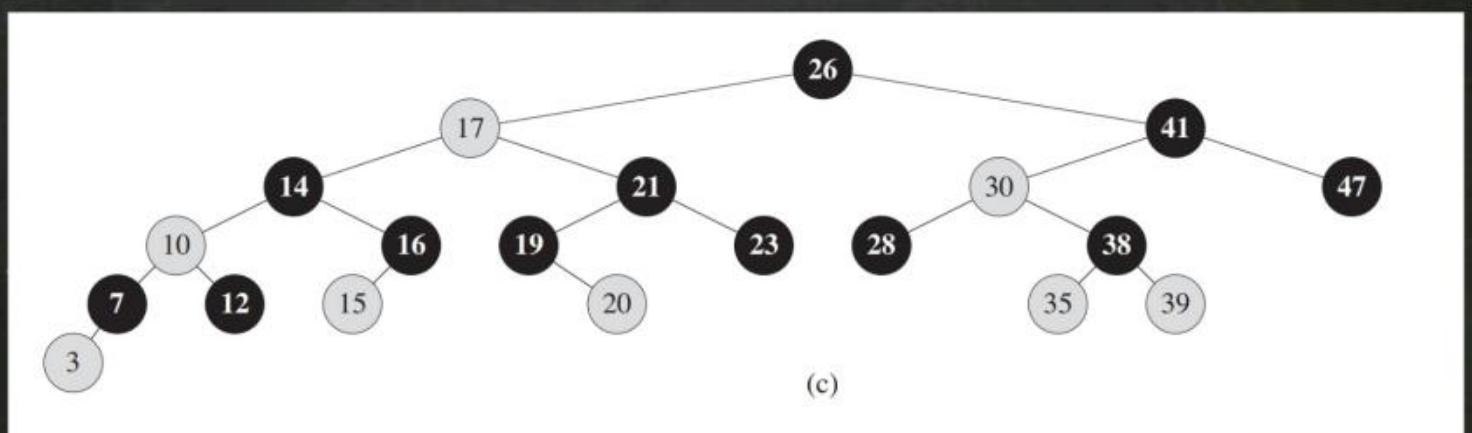
عصر تابان (گروه تابان) ← گره لوح
nil node مخصوص از طلاس
و خصیص حافظه داده‌ها



```
class RBtree {
    RBnode *root;
    RBnode *nil;
    ...
    ...
    RBtree() {
        nil=new RBnode
    }
}
```

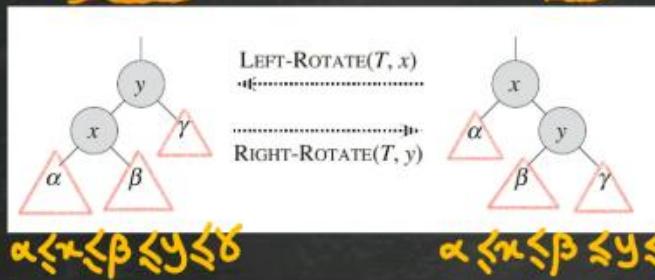
و کردار گنیم مقدار غلبه کننده برای گره nil را همچو ۵۰ در
اسناد و کار نمودیم و چنین صورت راهگردانی را داریم که در این می‌گذیم.
در واقع هر سعی از نوع RBtree برای خود یک گره nil دارد.

صرفهای سدی و خلوتی سهل از اینجا به بعد که nil را در صورت اس نمی‌فهم



عملکردهای چرخش

→ در رام ساختارهای درخت جستجوی موزن موردنظر قرار ندارند



برای عملکردی اینها برای تغییر موضعی ساختار ارتباطی تر می‌باشد.
درخت جستجوی دو دویی است که ممکن حفظ ویرایش
ارتفاع زید درخت را تغییر می‌دهد.

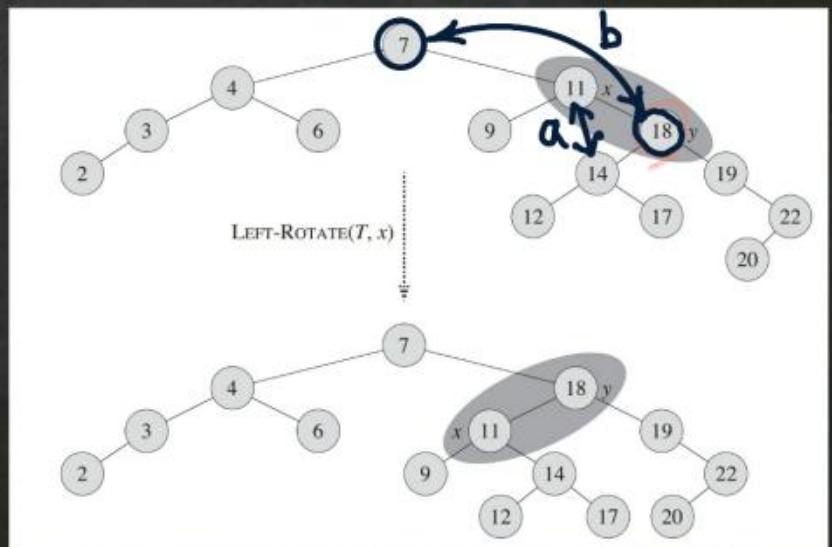
LEFT-ROTATE(T, x)

```

1  y = x.right
2  x.right = y.left
3  if y.left ≠ Nil
4    y.left.p = x
5  y.p = x.p
6  if x.p == T.nil
7    T.root = y
8  elseif x == x.p.left
9    x.p.left = y
10 else x.p.right = y
11 y.left = x
12 x.p = y

```

$O(1)$



۱۰) در درخت رزون-سیاه

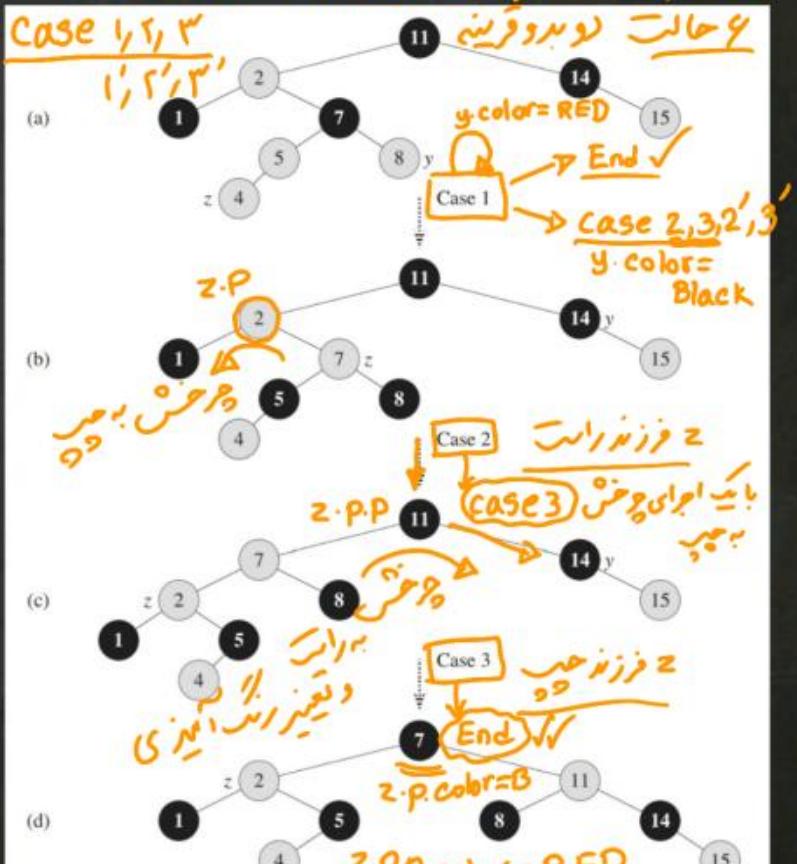
RB-INSERT(T, z)

```

1    $y = T.nil$ 
2    $x = T.root$ 
3   while  $x \neq T.nil$ 
4      $y = x$ 
5     if  $z.key < y.key$ 
6        $x = x.left$ 
7     else  $x = x.right$ 
8    $z.p = y$ 
9   if  $y == T.nil$ 
10     $T.root = z$ 
11  elseif  $z.key < y.key$ 
12     $y.left = z$ 
13  else  $y.right = z$ 
14   $z.left = T.nil$  ✓✓
15   $z.right = T.nil$ 
16   $z.color = RED$ 
17  RB-INSERT-FIXUP( $T, z$ )

```

حالت های مردمیار در RB-INSERT-FIXUP



RB-INSERT-FIXUP(T, z)

```
1  while  $z.p.color == \text{RED}$ 
2    if  $z.p == z.p.p.left$ 
3       $y = z.p.p.right$ 
4      if  $y.color == \text{RED}$ 
5         $z.p.color = \text{BLACK}$ 
6         $y.color = \text{BLACK}$ 
7         $z.p.p.color = \text{RED}$ 
8         $z = z.p.p$ 
9    else if  $z == z.p.right$ 
10       $z = z.p$ 
11      LEFT-ROTATE( $T, z$ )
12       $z.p.color = \text{BLACK}$ 
13       $z.p.p.color = \text{RED}$ 
14      RIGHT-ROTATE( $T, z.p.p$ )
15    else (same as then clause
16      with "right" and "left" exchanged)
17     $T.root.color = \text{BLACK}$ 
```

در این حالت

من اینجا همچو robe اینم

حالات دارم که مرا خواهی عرض کرد
حالت های عرض کرد

$y.color = \text{BLACK}$

// case 1
// case 1
// case 1
// case 1

// case 2
// case 2
// case 3
// case 3
// case 3