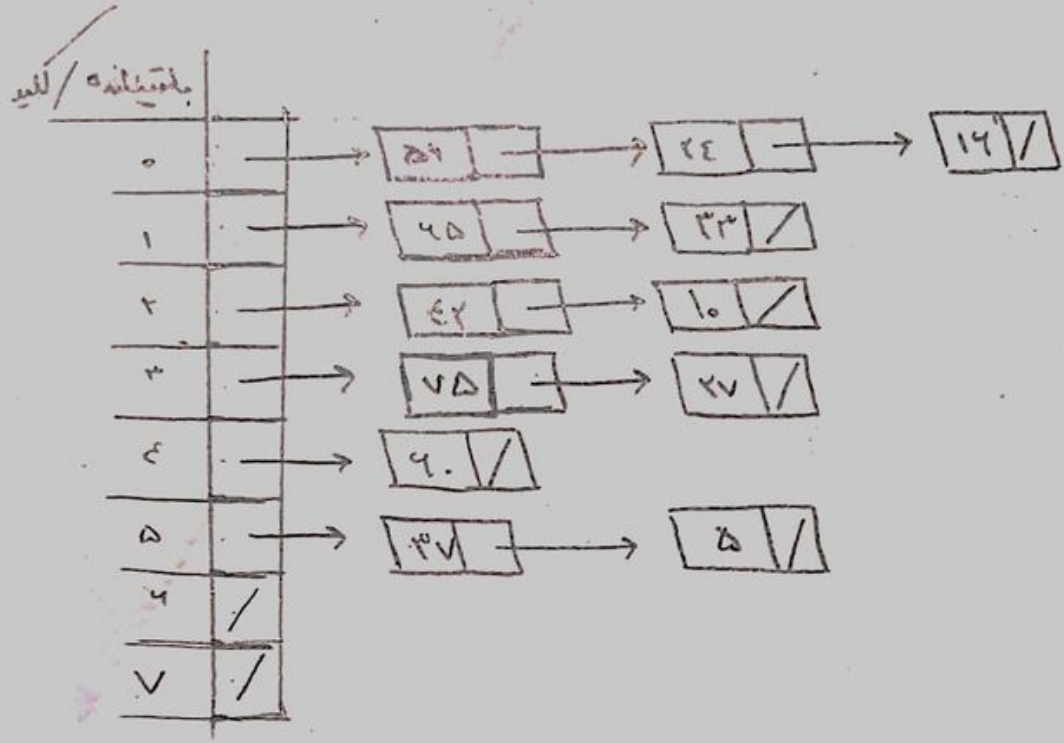


①

نمایش

با لیس پیوندی



نمایش جدولی ساده

مقدار	باقی‌مانده / کلید
۵۴, ۴۴, ۱۴	۰
۴۵, ۳۳	۱
۴۲, ۱۰	۲
۷۵, ۲۷	۳
۹۰	۴
۳۷, ۵	۵
∅	۶
∅	۷

۲۷

الف) ارتفاع منتهی گره : چونه ضلیدی نه حاجتی ایدم به تر اماند لینم نه همان black height

گره های گول با استناد به میلد بیمه های یعنی node.left.field و node.right.field

بیش آوردن طبق قضیه اوجا « « RB « گول میلد ارتفاع منتهی گره

را به آن augment لینم و هر عملیات Delete, Insert, Search را « مرتبه $O(\lg n)$ داشته باشیم.

ب) حق گره : دمه حق گره نام های Search, Insert, Delete گول

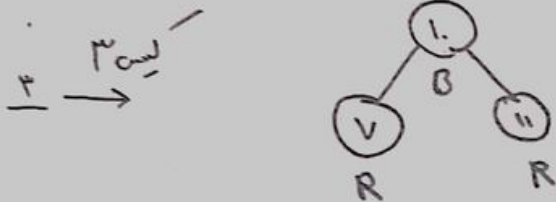
همیشه مال قبلی از مرتبه $O(\lg n)$ دات اما « مورد Delete چونه اسریشی

یک زیر درخت حذف شود، آناه بله کتا گره های آن زیر درخت را جدا عصمتان را اندازه بگیریم،

بیش اسریشی دمت حذف شود نام Delete از مرتبه $O(\lg n)$ می شود.

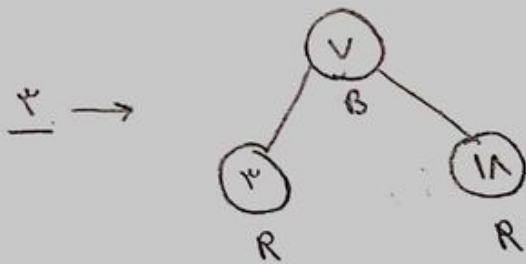
۳) از مثال نقص برای رد تمهید اسنادی کنیم، ابتدا به عدد ۱۱، ۱۰، ۷ را

به ترتیب insert می کنیم. لا راضی می کنیم به حالت قبل از insert(۷) می کنیم

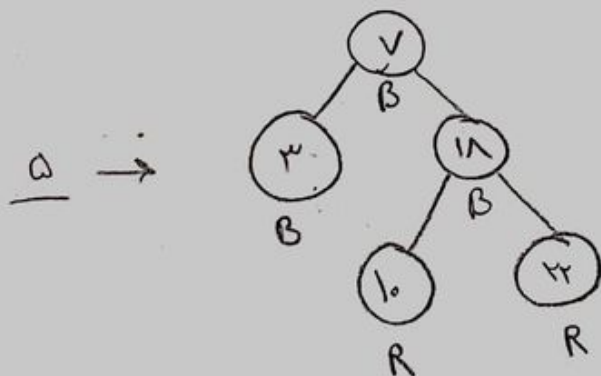
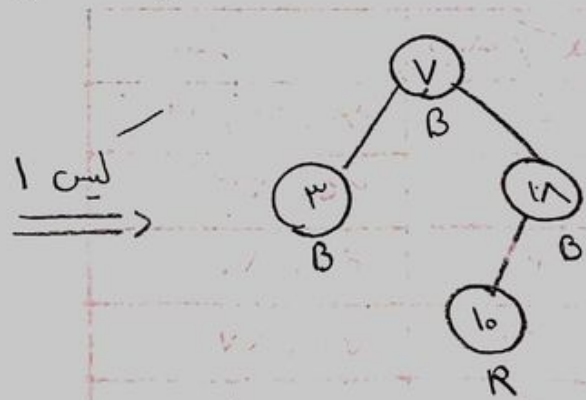
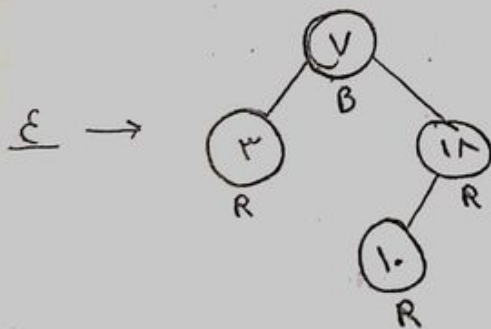


ملاحظه می کنیم حالت های ۴ و ۲ به دلیل نیت به سمت راستی و به سمت چپ RB اضافه و سپس حذف کنیم، به جهت ادره برتری داریم.

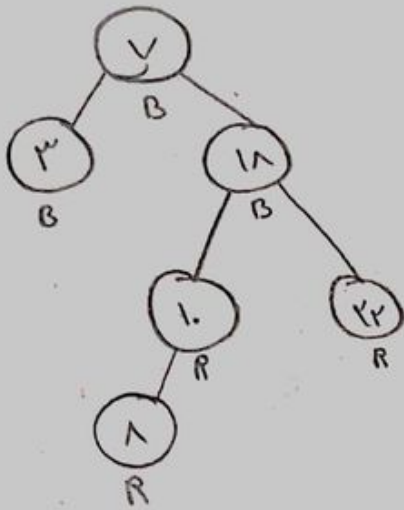
۴) برای ساری زنده بودگی را با B و دمنز را با R «پایین هر نو سازی
ی دهم (برو مارا هم سازی بن دهم) اگر امانه کرده را عبور بکن من غیر ادبی، قمری لیس



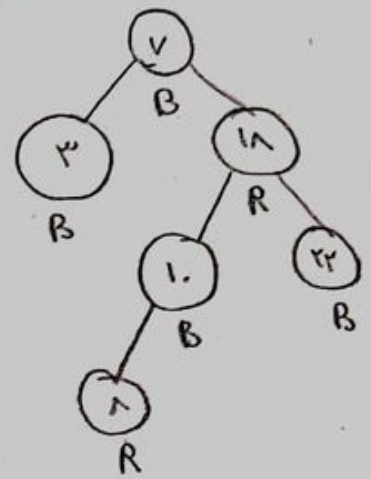
تا به مرحله اول مشکل لیس خاصی نداریم، ۷ به ریشه است مشکلی نبود و ۳، ۱۱ چون به های
چپ و راست ریشه هستند، برای هم ترتیب آرایش مشکلی ریشه قمری شوند.



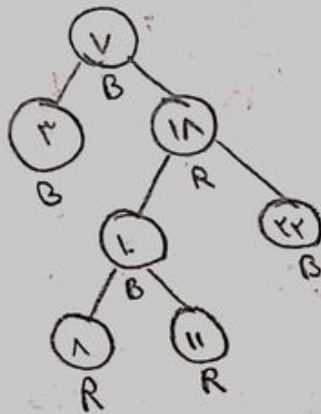
9 →



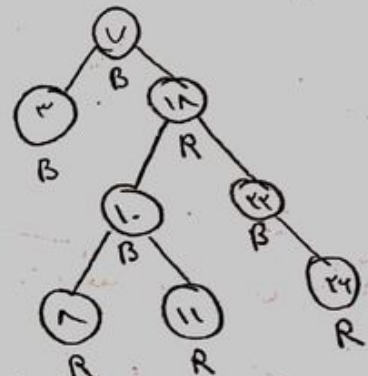
لست
⇒



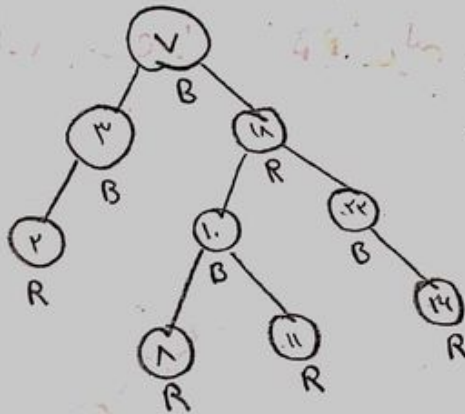
17 →



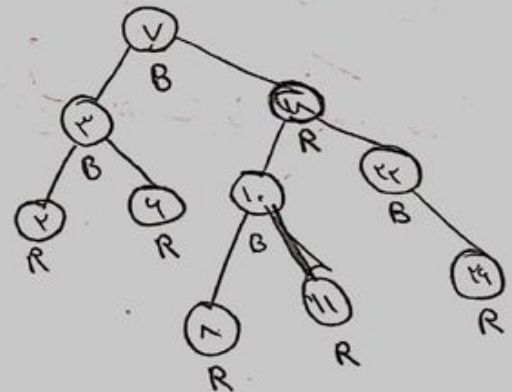
17 →



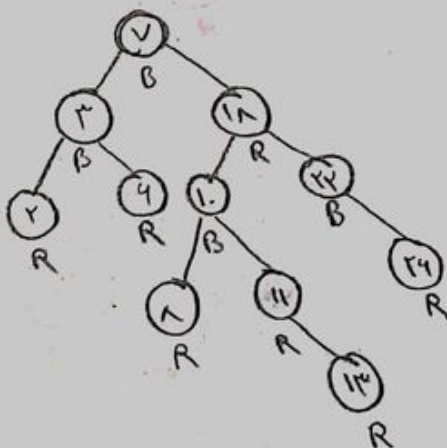
9 →



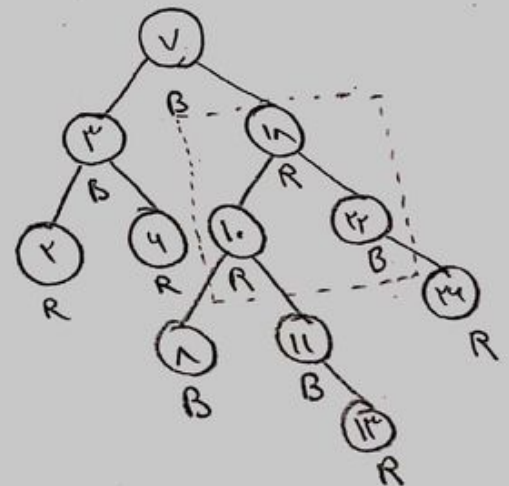
17 →



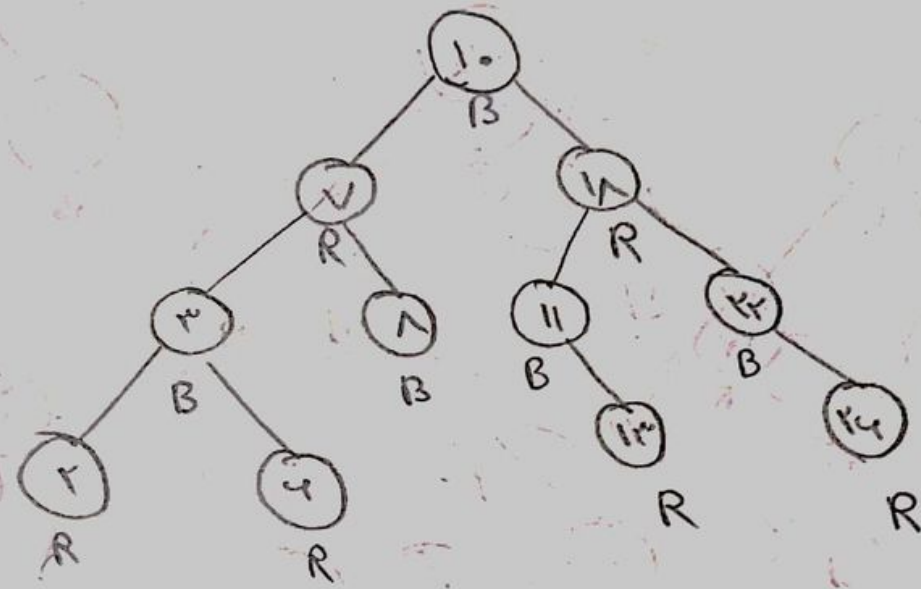
11 →



لست
⇒



لیست
==>



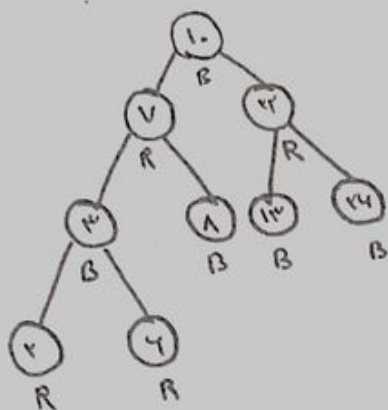
* در هر مرحله طبق insert + BST ، نو دهارا اضافه می کنیم لیست ریشه آن را
 به طور پیش فرض فرض می کنیم ، اگر بعد از insert ، والد نو دهارا پیدا نکند ، نو دهارا لیست جدید
 به با اعداد مثل آن در می آید . در هر insert هم ریشه درخت را سیاه می کنیم .

RB-INSERT-FIXUP(T, z)

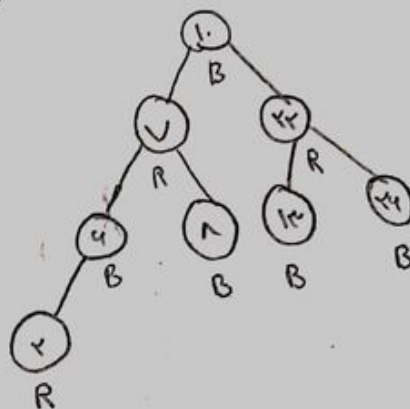
```
1  while  $z.p.color = RED$ 
2      if  $z.p = z.p.p.left$ 
3           $y = z.p.p.right$ 
4          if  $y.color = RED$ 
5               $z.p.color = BLACK$            // Case 1
6               $y.color = BLACK$            // Case 1
7               $z.p.p.color = RED$          // Case 1
8               $z = z.p.p$                  // Case 1
9      else if  $z = z.p.right$ 
```

```
10          z = z.p                                // Case 2
11          LEFT-ROTATE(T, z)                        // Case 2
12          z.p.color = BLACK                          // Case 3
13          z.p.p.color = RED                          // Case 3
14          RIGHT-ROTATE(T, z.p.p)                    // Case 3
15      else (same as then clause with "right" and "left" exchanged)
16  T.root.color = BLACK
```

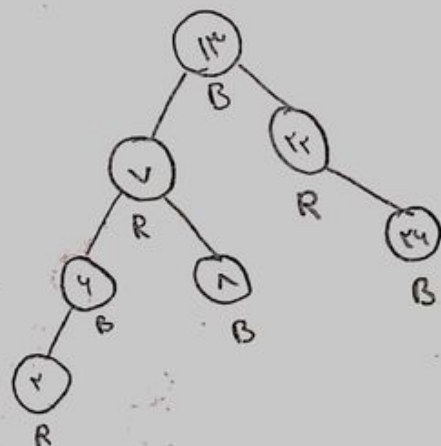

۲ →



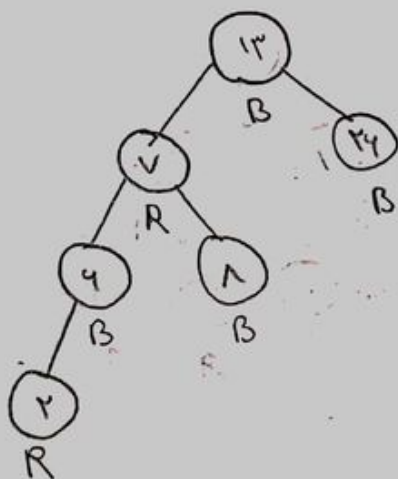
۲ →



۳ →



۵ →



* «مرحله ماته delete» BST «در نیاز از + transplan اسناد می کنیم
و «دست» BST را می کنیم و به با اسناد از «نیم» موجود می کنیم «دست» RB
نصف شده باشد.

```

4      if w.color == RED
5          w.color = BLACK // case 1
6          x.p.color = RED // case 1
7          LEFT-ROTATE(T, x.p) // case 1
8          w = x.p.right // case 1
9      if w.left.color == BLACK and w.right.color == BLACK
10         w.color = RED // case 2
11         x = x.p // case 2
12     else if w.right.color == BLACK
13         w.left.color = BLACK // case 3
14         w.color = RED // case 3
15         RIGHT-ROTATE(T, w) // case 3
16         w = x.p.right // case 3
17         w.color = x.p.color // case 4
18         x.p.color = BLACK // case 4
19         w.right.color = BLACK // case 4
20         LEFT-ROTATE(T, x.p) // case 4
21         x = T.root // case 4
22     else (same as then clause with "right" and "left" exchanged)
23     x.color = BLACK

```