

قصر ۴ سافتمان دله هاو الگورتم ها

اِسْکَانِ شِکِلِیَا

9931.15

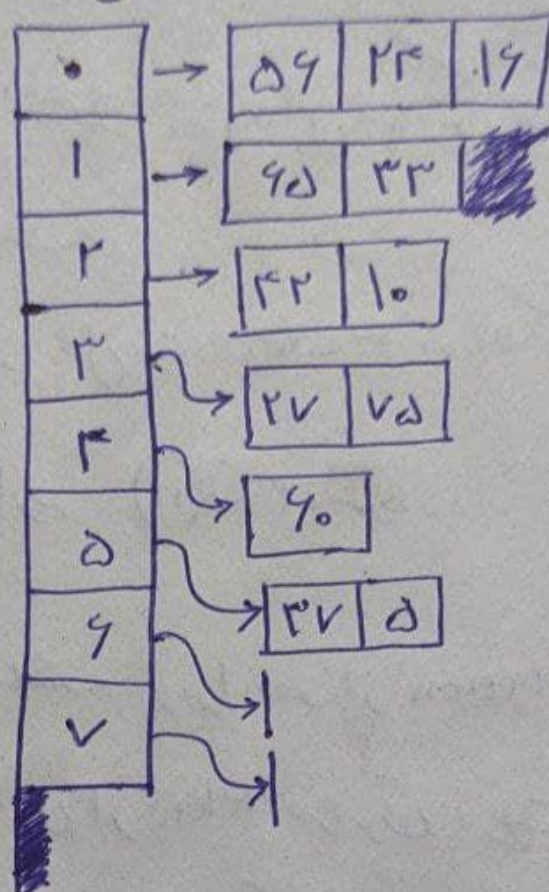
(1)

$$90 \rightarrow 1$$
$$\Delta \eta \rightarrow 0$$
$$\mathbb{R}^2 \rightarrow \mathbb{R}$$
$$10 \rightarrow 2$$
$$y_0 \rightarrow \infty$$
$$2F \rightarrow 0$$
$$33 \rightarrow 1$$
$$V\Delta \rightarrow \mathbb{R}$$
$$\Gamma \rightarrow \Delta$$

19 → 0

$$2V \rightarrow 3$$
$$\Delta \rightarrow \omega$$

Array



(۲) ارتعاش مٹکی کرو و عمق کرو:

Search : با توجه به اینکه درخت همپایان یک BST است، پس می‌توان با این سری سرچ انجام داد که همان $O(\log n)$ است و ربطی به ارتفاع مشکلی و عموی نه‌ها ندارد.

insert: با سرچ محل خود را یافته و خود را به شکل قمر اضافه کرده و fix می کنیم.
برای ارتعاشی
و سپس با بررسی ارتعاشی گری فرزندان خود جدید، رنگ آن را تعیین می کنیم.

برای عمق بعد از fix up و با بررسی عمق فرزندان، عمق خود را می‌سپارد و

پس در ارتفاع مشکلی، ~~بسیار~~ بیخبر کسی بغیر نمی کند اما در عمق کره ~~بسیار~~

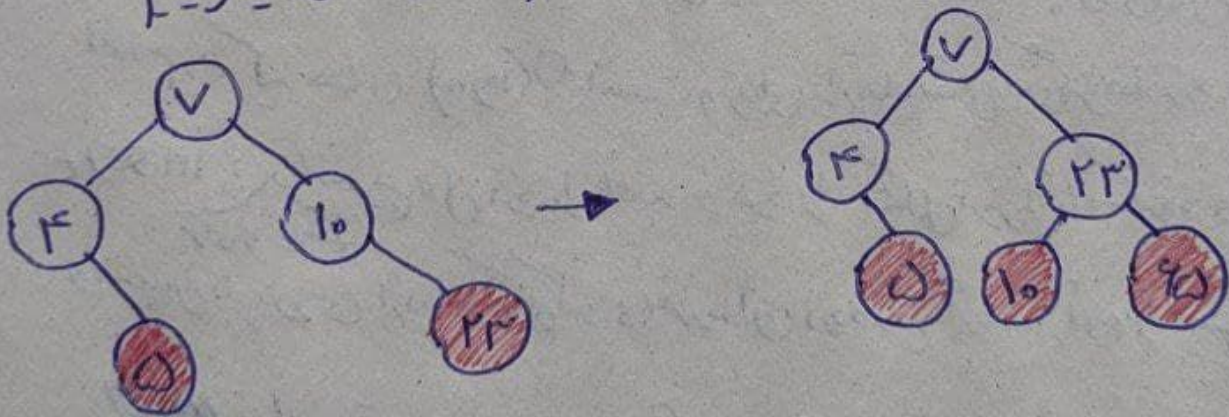
اردر $O(n)$ شود.

delete : برای ارتفاع مشکلی، اگر از محل نود حذف شونده پایین برویم و ارتفاع مشکلی گره‌ها را تغییری دهیم، این پیمایش که *traversal* نیست، از اردر $O(\log n)$ است.

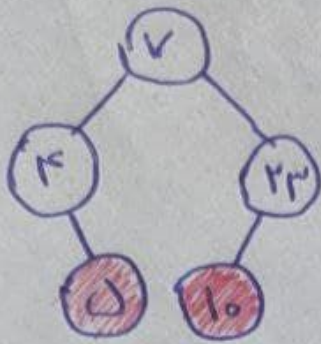
برای عمق، چون از محل *predecessor* به بالا حرکت می‌کنیم تا عمق گره‌ها را تغییر دهیم، از اردر $O(n)$ می‌شود.

(۳) ممکن است یکسان نباشد، زیرا بعد از *rotation* در عمل *insert* درخت شکلی می‌گیرد که پیش از *delete* در صورتی درخت اولیه حاصل می‌شود که خلاف *rotation* گفته شده انجام شود که لزوماً اینطور نیست.

مثال نقض : درخت با اعضای ۴، ۷، ۱۰، ۲۳ و ۵ که به همین ترتیب *insert* شده‌اند در نظر بگیرید که تلاش می‌کنیم ۶۵ را به آن بیفزاییم :

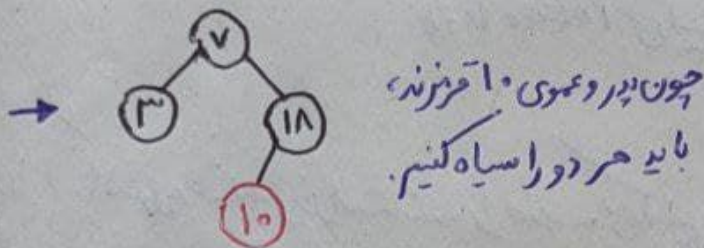
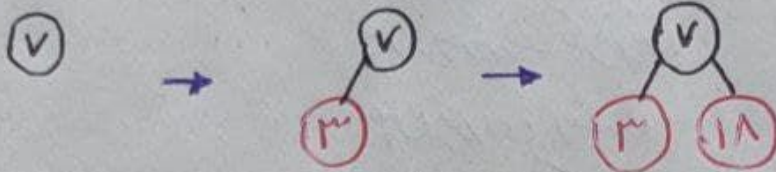


حال با حذف گره ۹۵ خواهیم داشت :

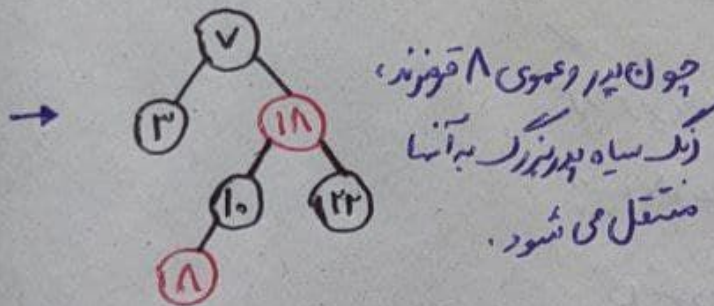
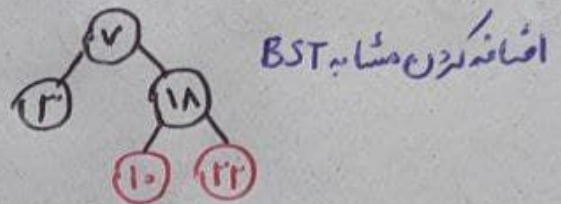


که متفاوت با درخت اولیه است.

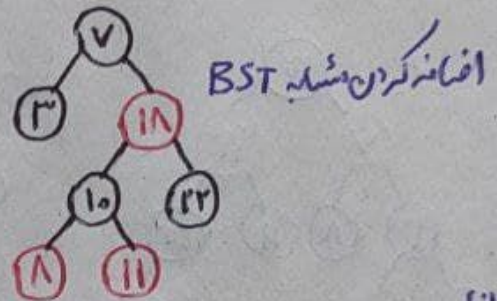
(۳) اضافه کردن مشابه BST
رشته سیاه و فرزندان قرمز



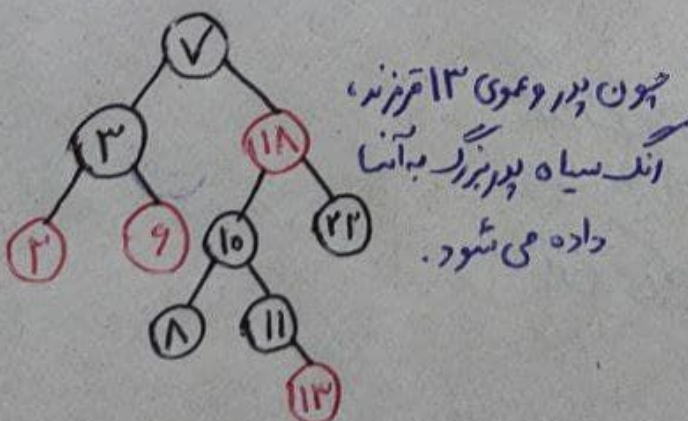
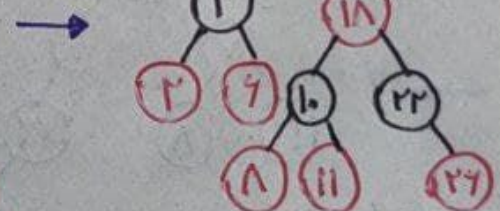
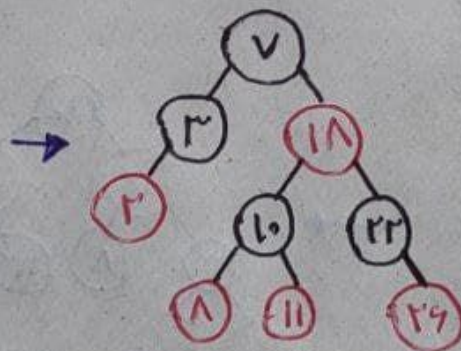
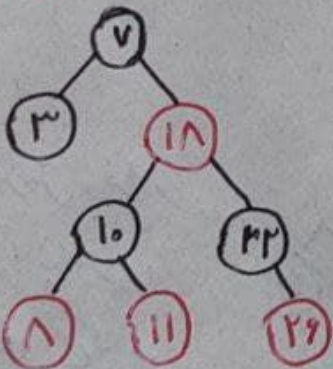
چون پدر و عموی ۱۰ فرزندان
باید هر دو را سیاه کنیم.



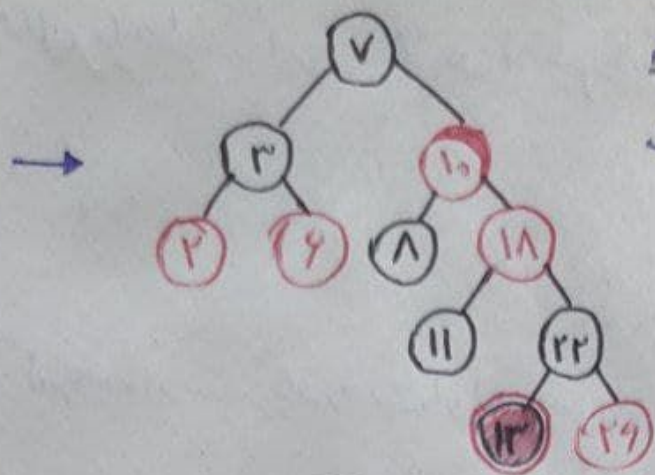
چون پدر و عموی ۸ فرزندان
رنگ سیاه پدر بزرگ به آنها
منتقل می شود.



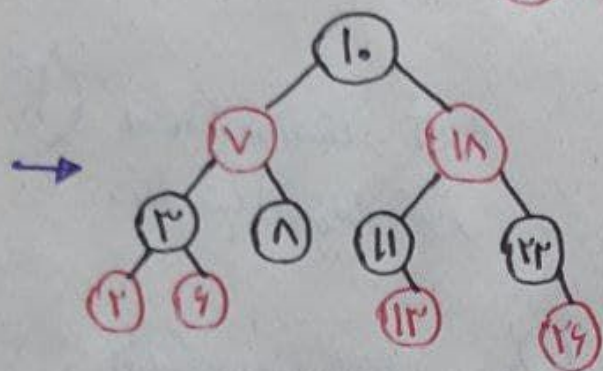
افزودن
مشابه BST



چون پدر و عموی ۱۳ فرزندان
رنگ سیاه پدر بزرگ به آنها
داده می شود.

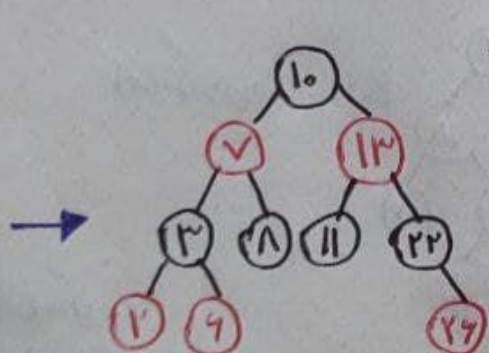


پدر و پسر بزرگ ۸ و ۱۱ قهرمانند؟
چون پدر چپ و پسر بزرگ راست
است rotate right

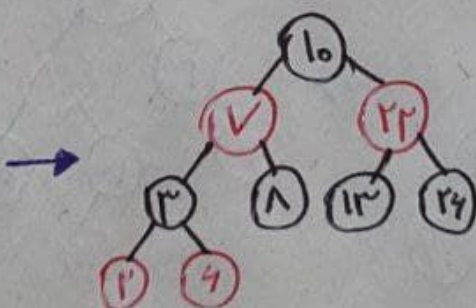
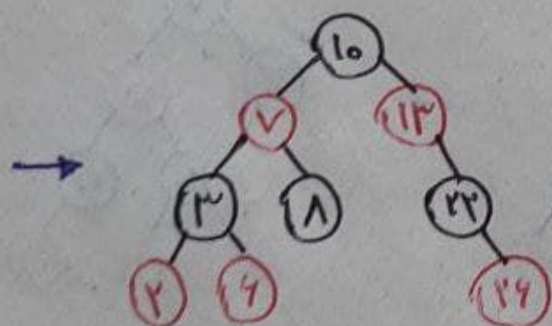


پدر و پسر بزرگ قهرمانند؟
پس برادی ریشه
rotate left می کنیم

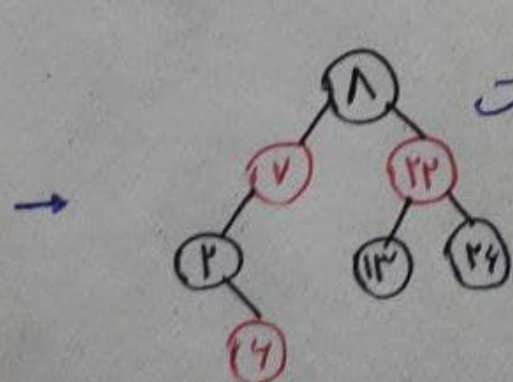
الکون عناصر خواسته شده را حذف می کنیم



بزرگترین عنصر زیر درخت
چپ ۱۸ جایگزین آن
می شود

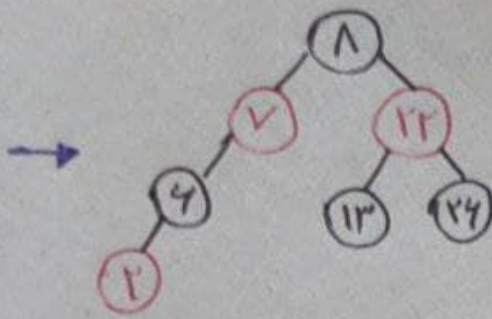


برادر زاده قهرمان و برادر
سیاه است؟ بنابراین
rotate left می کنیم

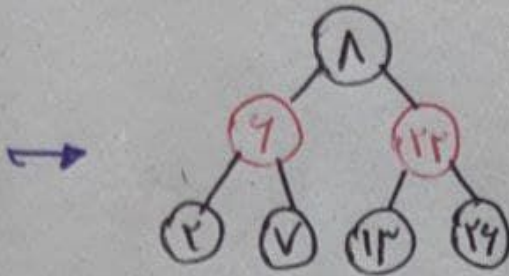


بزرگترین عنصر زیر درخت
چپ ۱۵ را جایگزین
آن می کنیم

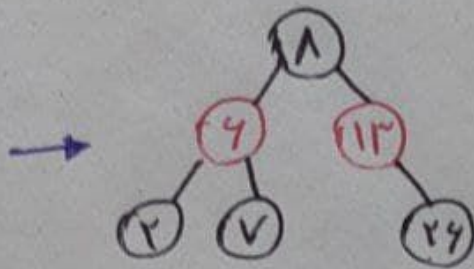
برادرها سیاه اند و برادرزاده
قرمز، پس بر روی ۲
rotate left می کنیم



برادرها سیاه اند و برادرزاده
قرمز، پس بر روی 6
rotate right می کنیم



بزرگترین مقدار زیر درخت
چپ ۲۲ را جایگزین
آن می کنیم.



سیاه های برادر یعنی باید
سیاه را یک مرحله
بالا تر ببریم.

