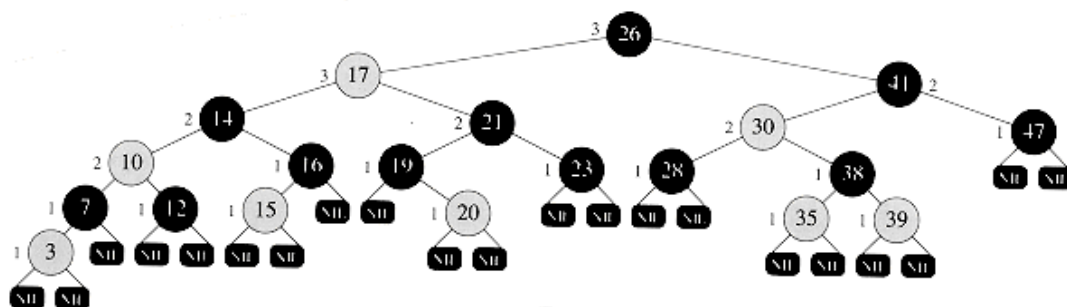


درخت قرمز-سیاه

درخت قرمز-سیاه یک درخت جستجوی دودویی است که با اختصاص دادن یک رنگ قرمز یا سیاه به هر گره شرایطی ایجاد می کنیم که ارتفاع درخت $O(\lg n)$ شود. بنابراین هزینه درج، حذف، جستجو و غیره که در درخت های BST به ارتفاع درخت بستگی دارد در زمان $O(\lg n)$ انجام می شود. با انجام اعمال درج و حذف، ارتفاع درخت از حالت لگاریتمی خارج نمی شود. البته در برخی حالات نیاز است که با انجام اعمالی، ارتفاع درخت را متوازن نگه داریم.

خواص درخت قرمز-سیاه:

- BST است.
- هر گره دارای یک مولفه رنگ سیاه یا قرمز است.
- فرزند هر گره قرمز حتما سیاه است.
- تعداد گره های سیاه در تمام مسیرها از گره x تا هر یک از برگ های نواده اش، برابر هستند که به این تعداد **black-height** گوئیم و آن را با $bh(x)$ نشان می دهیم. (رنگ خود x را نمی شماریم).
- تمام برگ ها، NIL (تهی) هستند و رنگ آن ها سیاه است. (برگ NIL برگی است که data ندارد)
- رنگ ریشه سیاه فرض می شود هر چند این شرط ضروری نیست.



یک درخت قرمز-سیاه با 20 گره داخلی که کلید هر گره در داخل آن و bh هر گره در کنارش نوشته شده.

قضیه : حداکثر ارتفاع یک درخت قرمز-سیاه که دارای n گره داخلی است برابر $2\lg(n+1)$ است.

اثبات: برای اثبات این قضیه ابتدا نشان می دهیم که هر زیردرخت دلخواه به ریشه x حداقل دارای $2^{bh(x)} - 1$ گره داخلی است. از استقرا روی ارتفاع x استفاده می کنیم:

پایه : اگر ارتفاع x صفر باشد، x برگ است و هیچ گره داخلی وجود ندارد بنابراین $2^{bh(0)} - 1 = 0$

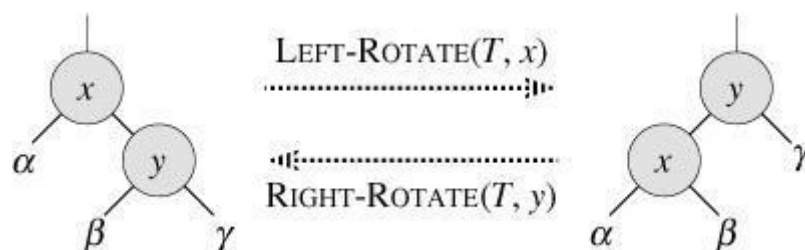
گام استقرا: bh فرزندان x ، با توجه به سیاه یا قرمز بودن x ، برابر $bh(x)$ یا $bh(x)-1$ است. از فرض استقرا نتیجه می گیریم زیردرخت های به ریشه ی هر فرزند x هر کدام حداقل $2^{bh(x)-1} - 1$ گره داخلی دارد. بنابراین زیردرخت به ریشه x حداقل $2(2^{bh(x)-1} - 1) + 1$ گره داخلی دارد.

در یک درخت به ارتفاع h حداقل نصف گره های هر مسیر از ریشه به برگ سیاه هستند، در غیر این صورت یک گره قرمز وجود خواهد داشت که فرزندش قرمز است که این ممکن نیست. در نتیجه bh ریشه حداقل برابر $h/2$ است. بنابراین :

$$2^{h/2} - 1 \leq n \rightarrow 2^{h/2} \leq n+1 \rightarrow h \leq 2\lg(n+1)$$

دوران

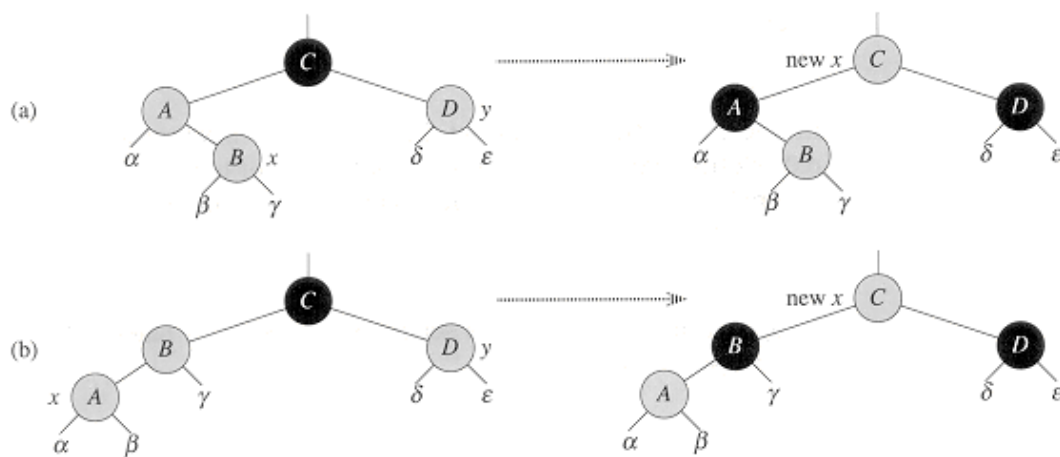
اگر در درخت قرمز-سیاه اعمال درج و حذف را انجام دهیم ممکن درخت به صورتی تغییر کند که دیگر خصوصیات درخت قرمز-سیاه را حفظ نکند و آن را نامتوازن کند. برای حل این مشکل از عمل دوران استفاده می کنیم. دوران یا راست گرد (right-rotate) است یا چپ گرد (left-rotate). اگر طبق شکل زیر دوران انجام دهیم رابطه $\alpha < x < \beta < y < \gamma$ حفظ می شود. (α ، β و γ زیردرخت هستند).



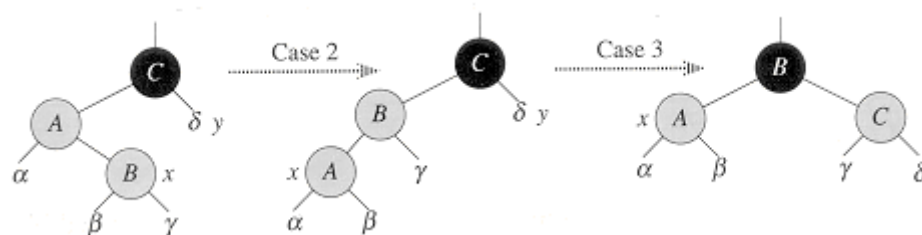
درج یک عنصر

ابتدا عنصر x را به همان صورت که در BST معمولی درج می کردیم، در درخت درج می کنیم و رنگ آن را قرمز می کنیم؛ اگر پدر x سیاه بود، درج به پایان می رسد. اگر پدر x قرمز باشد:

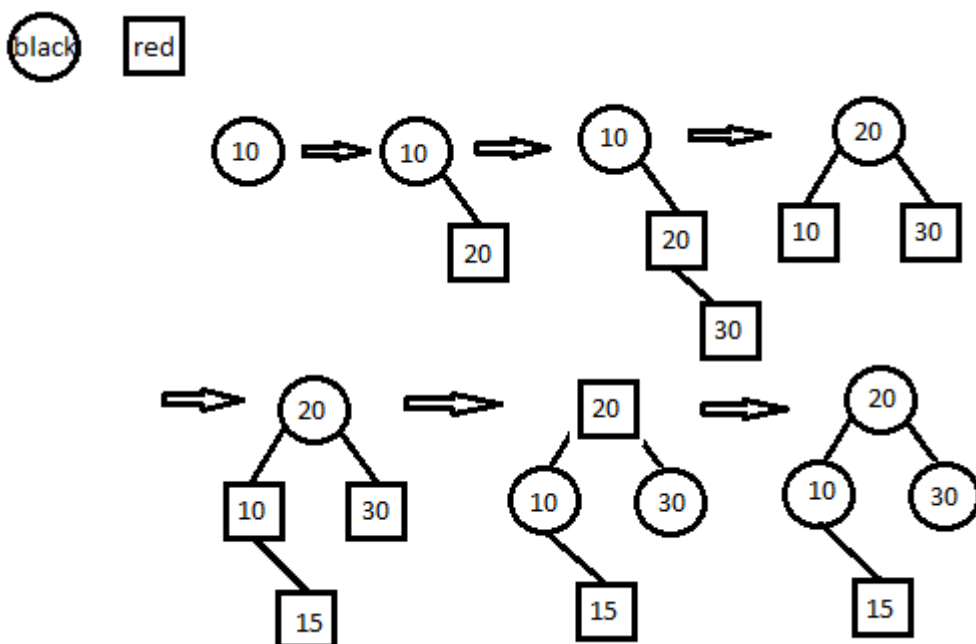
الف) اگر عمومی x به نام y قرمز باشد، پدر y با برچسب C حتما سیاه است. در این صورت رنگ C را به قرمز و رنگ y و پدر x را به سیاه تغییر می دهیم. اگر پدر C قرمز باشد این الگوریتم را به صورت بازگشتی روی $\text{new } x$ اجرا می کنیم.



ب) اگر y سیاه باشد؛ اگر x فرزند راست پدرش باشد، با یک دوران چپ گرد بین x و پدرش، آن را به حالتی تبدیل می کنیم که فرزند چپ پدرش شود. سپس بین B و C یک دوران چپ گرد انجام می دهیم و رنگ آن ها را تغییر می دهیم.



مثال: درج اعداد 10، 20، 30 و 15 در یک درخت خالی:



حذف یک عنصر

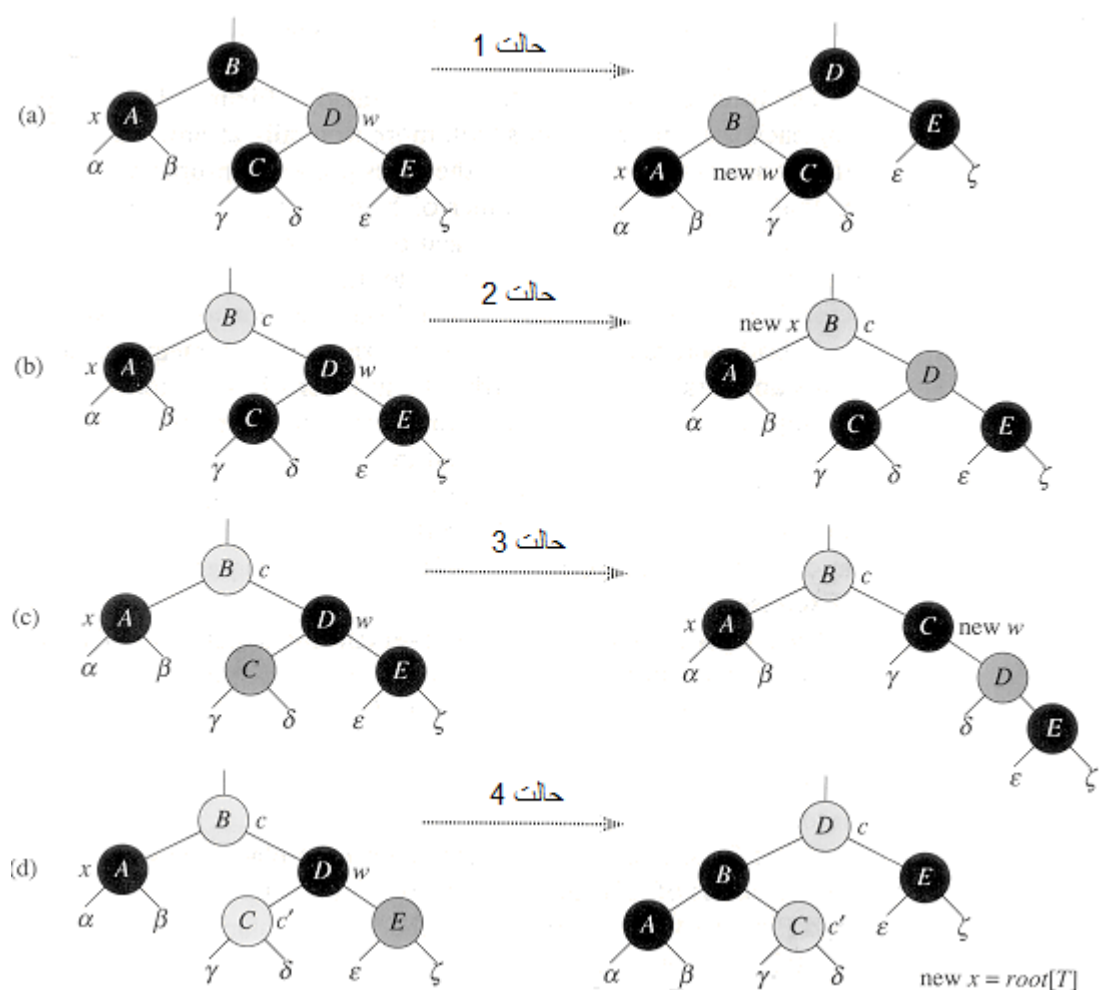
عنصر مورد نظر Z را مانند الگوریتم حذف در BST ساده، حذف می کنیم که در آن یک گره Y به صورت فیزیکی از درخت حذف می شود که می تواند همان Z باشد یا می تواند یک گره دیگر باشد. اگر Y فرزند غیرتهی داشته باشد X فرزند غیرتهی برای Y است در غیر این صورت X را یک فرزند تهی Y در نظر می گیریم. اگر رنگ Y قبل از حذف قرمز بوده باشد، حذف آن مشکلی ایجاد نمی کند ولی اگر رنگ Y سیاه باشد؛ فرض می کنیم که X علاوه بر رنگ خود یک رنگ سیاه اضافی هم دارد. با این فرض همه bh ها درست می شوند. حال باید این رنگ سیاه اضافی را به سمت ریشه هدایت کنیم تا درنهایت یا جایگزین یک رنگ قرمز شود یا به ریشه برسد و دور ریخته شود. اگر برادر X، W باشد، 4 حالت پیش می آید:

1. W قرمز است (بنابر این فرزندانش سیاه اند) : در این حالت با تعویض رنگ گره های B و D و انجام یک دوران چپ گرد و با نام گذاری برادر جدید X، مسأله را به یکی از حالات 2، 3 یا 4 تبدیل می کنیم.
2. W و دوفرزندش سیاه اند : W را قرمز می کنیم و رنگ سیاه اضافی X را به پدرش که در حال حاضر رنگ C را دارد منتقل می کنیم. اگر C قرمز باشد به سیاه تغییر رنگ می دهد و

عملیات حذف تمام می شود. اگر x ریشه باشد هم عملیات پایان می یابد، در غیر این صورت الگوریتم را به همین شکل ادامه می دهیم.

3. w سیاه است و فرزند راستش سیاه و فرزند چپش قرمز است : این حالت را با تعویض رنگ گره های C و D و انجام یک دوران راستگرد به حالت 4 می رسیم.

4. w سیاه است و فرزند راستش قرمز است : در این حالت با تغییر چند رنگ و یک دوران چپ گرد، رنگ سیاه اضافی x را حذف می کنیم.



گره ها یا سیاه اند یا قرمز یا دارای رنگ c یا c' هستند.