ساختمان داده‌ها (۲۲۸۲۲)

مدرس: حسین بومری

[بهار ۹۹]

جلسه ۱۹: آشنایی با درخت های دودویی جستجو متوازن

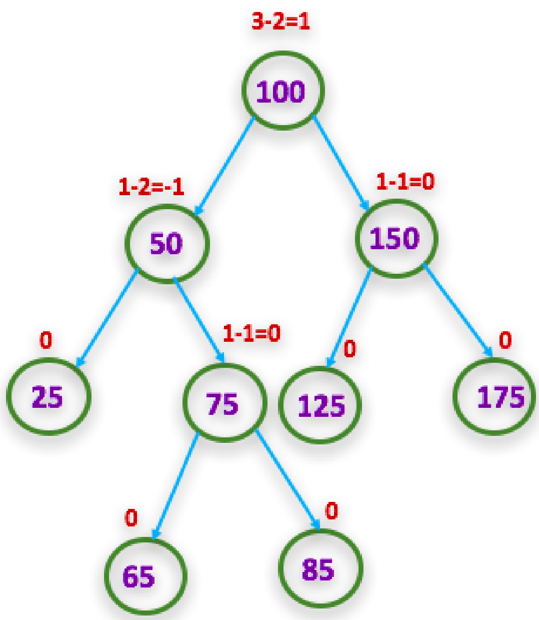
نگارنده: مبین معدنی

در جلسات پیش با تعریف توابع مورد نیاز برای درخت ها دودویی متوجه این شدیم که ارتفاع درخت ها در اکثر توابع تاثیر بسزایی دارد و هر چه ارتفاع درخت ما کمتر باشد ، مرتبه زمان انجام توابع گفته شده نیز کمتر میشود ، حال در این جلسه قرار است با دو شیوه برای متوازن سازی درخت ها دودویی آشنا شویم که کمک میکند ارتفاع درخت های ما بهینه شود.

درخت های متوازن :

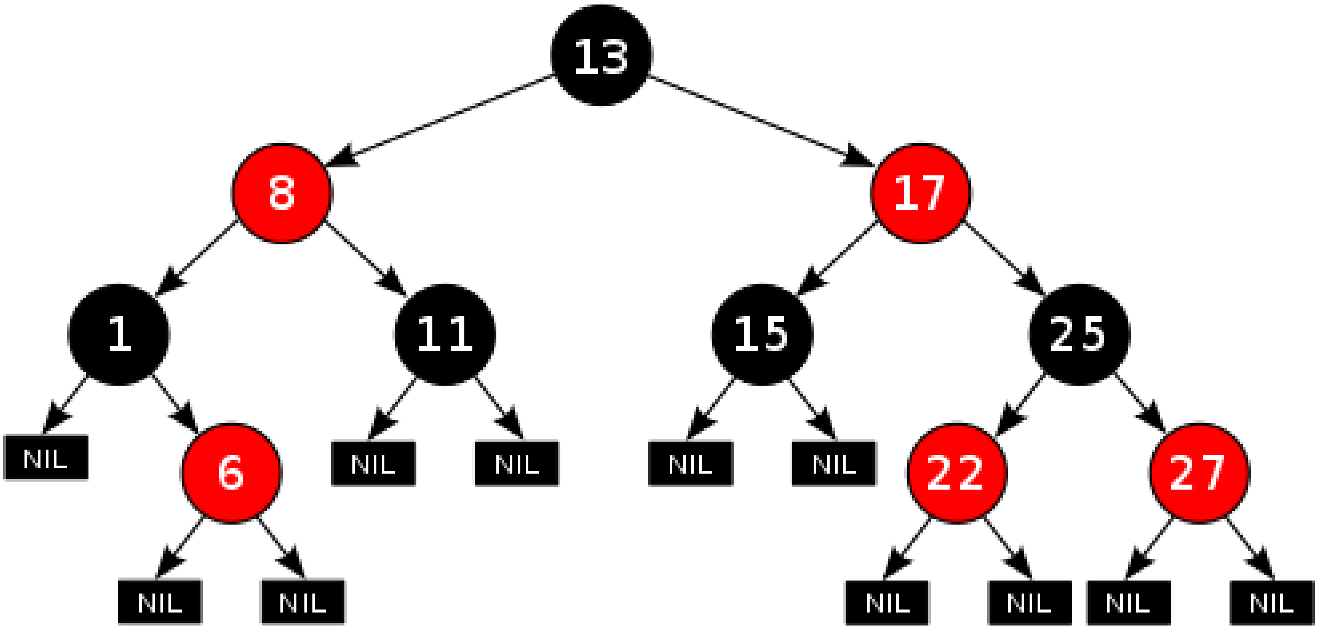
به ازای هر عنصر یک متغیری نگه داری میشود که نشان دهنده ی اختلاف عمق سمت چپ و راست آن عنصر است و دارای مقادیر $-۱, ۰, ۱$ است. و همیشه سعی میکنیم که اختلاف عمق ها را در این بازه حفظ کنیم.

۱.۰ درخت های AVL



AVL Tree

۲.۰ درخت های قرمز مشکی



هر نود دارای رنگی مشخص است (مشکی ، قرمز) و درخت از قواعد زیر پیروی میکند.

برگ ها همیشه مشکی هستند.

تعداد خونه های مشکی دیده شده از هر عنصر تا فرزند های null باید مقدار برابر باشد.

دو قرمز پشت هم نمیتوانند قرار داشته باشند.

بررسی درخت AVL

حال بررسی میکنیم که در یک درخت با عمق H حداقل چند عنصر قرار خواهد داشت:

$$T(H) = T(H-۱) + T(H-۲) + ۱$$

حداقل تعداد عنصر زمانی رخ میدهد برابر است با حداقل عنصر های یکی از فرزند هایش "$T(H-۱)$"و از آنحا که اختلاف عمق دو طرف میتواند حداکثر ۱ باشد فرزند دیگر دارای عمق $H-۲$ میشود پس $T(H-۲)$به این علت است. تابع بازگشتی ذکر شده مانند دنباله ی فیبوناتچی است پس مقدار آن برابر :

$$T(H) = (\frac{۱+\sqrt{۵}}{۲})^H$$

با توجه به نتیجه ی گفته شده میتوان نتیجه گرفت که اگر nعنصر داشته باشیم ، حداکثر

$$\log_{\frac{۱+\sqrt{۵}}{۲}}(n)$$

در جلسه ی بعد روش های حذف و اضافه کردن هر عنصر را در درخت های گفته شده بررسی خواهیم کرد