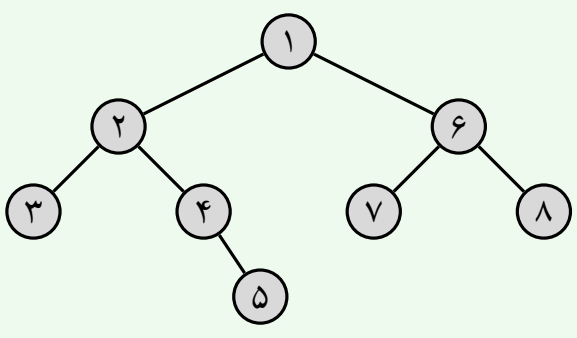
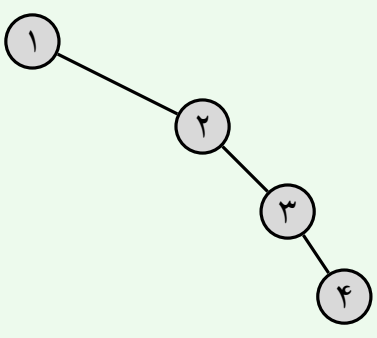
یادآوری جلسه چهاردهم — درخت‌های دودویی جستجوی متوازن — درّین ستوده

در جلسه گذشته به داده ساختارهای ددج متوازن پرداختیم و درخت AVL را مورد بررسی قرار دادیم. ابتدا به تعریف درخت متوازن می‌پردازیم:

درخت متوازن، درختی است که ارتفاعی از مرتبه $\mathcal{O}(logn)$ داشته باشد(n تعداد راس‌های درخت است). درخت قرمز_سیاه، درخت ۲ـ۴ و درخت AVL نمونه‌هایی از درخت ددج متوازن می‌باشند.



شکل ۱: درخت دودویی متوازن



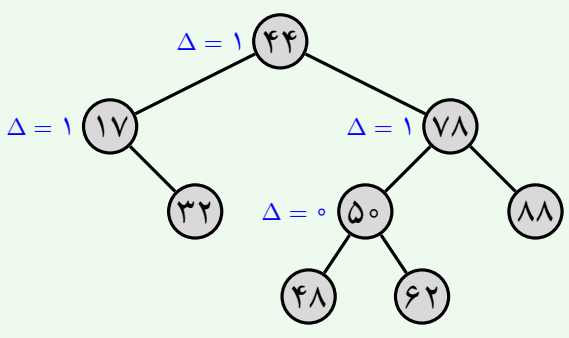
شکل ۲: درخت دودویی نامتوازن

اکنون به بررسی درخت AVL می‌پردازیم:

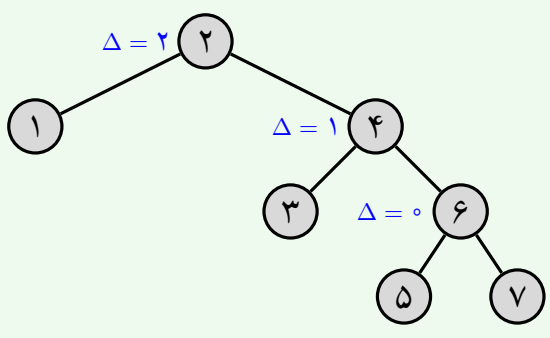
درخت AVL، یک درخت دودویی جستجو است که به ازای هر راس v در این درخت، ارتفاع زیردرخت چپ (h_{l_v}) و راست (h_{r_v}) آن حداکثر یک واحد اختلاف دارند.

$$\Delta_v = |h_{l_v} - h_{r_v}|$$

مثال: می‌خواهیم AVL بودن هر یک از درخت‌های زیر را بررسی کنیم.



شکل ۳: $AVL \checkmark \Leftarrow BST \checkmark,\ \forall v : \Delta_v \leq ۱$

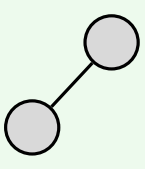


شکل ۴: $AVL \times \Leftarrow BST \checkmark, \exists v : \Delta_v > ۱$

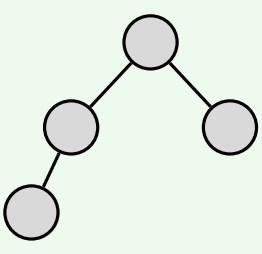
اثبات متوازن بودن درخت AVL:

$f(k)$ را حداقل تعداد راس در یک درخت AVL به ارتفاع k در نظر می‌گیریم.

شکل ۵: $f(۱) = ۱$



شکل ۶: $f(۲) = ۲$



شکل ۷: $f(۳) = ۴$

درخت T با k راس درنظر بگیرید. a را ریشه T، b و c را به‌ترتیب فرزندان چپ و راست آن می‌نامیم.

$$h_a = k \rightarrow max(h_b, h_c) = k-۱ \xrightarrow{\text{فرض میکنیم ارتفاع زیردرخت چپ بیشتر است}} \begin{cases} min(h_b) = k-۱ \\ min(h_c) = k-۲ \end{cases} \rightarrow f(k) = f(k-۱) + f(k-۲) + ۱$$

$$\rightarrow f(k) \geq \underbrace{f(k-۱) + f(k-۲)}_{\text{فیبوناچی}} = \phi^k \quad (\phi = \text{نسبت طلایی} \approx ۱/۶۲)$$

بنابراین اگر درخت AVL، n راس داشته باشد، ارتفاع آن از مرتبه $\mathcal{O}(logn)$ است. □

عملیات درج در درخت AVL:

برای درج یک عنصر در یک درخت AVL به‌صورت زیر عمل می‌کنیم:

۱. عدد را با استفاده از الگوریتم درج در درخت‌های BST، درج می‌کنیم.

۲. پایین‌ترین عنصری که شرط توازن در آن نقض می‌شود را می‌یابیم و آنرا z می‌نامیم.

۳. باتوجه به موقعیت عنصر اضافه شده نسبت به z، یکی از چرخش‌های زیر را به گونه‌ای انجام می‌دهیم تا شرط کوچکتر بودن هر فرزند چپ و بزرگتر بودن هر فرزند راست از پدرش برای زیردرخت حاصل برقرار باشد.

(آ) **چرخش LL:** برای حالتی که عنصر اضافه شده در زیردرخت چپ فرزند چپ عنصر z قرار گرفت.

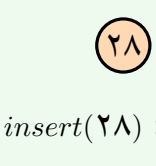
(ب) **چرخش LR:** برای حالتی که عنصر اضافه شده در زیردرخت راست فرزند چپ عنصر z قرار گرفت.

(ج) **چرخش RL:** برای حالتی که عنصر اضافه شده در زیردرخت چپ فرزند راست عنصر z قرار گرفت.

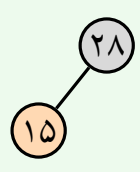
(د) **چرخش RR:** برای حالتی که عنصر اضافه شده در زیردرخت راست فرزند راست عنصر z قرار گرفت.

مثال: مراحل درج به ترتیب عناصر زیر در درخت AVL، در شکل‌های زیر آمده است.

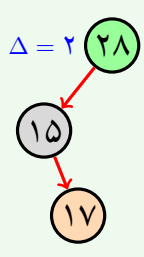
$$\{۲۸، ۱۵، ۱۷، ۴۸، ۵۰، ۲۱، ۱۱، ۵، ۱۶\}$$



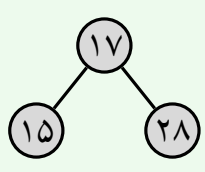
شکل ۸: $insert(۲۸)$



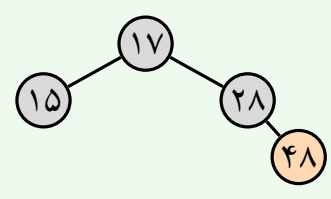
شکل ۹: $insert(۱۵)$



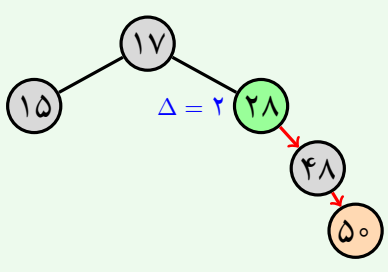
شکل ۱۰: $insert(۱۷)$



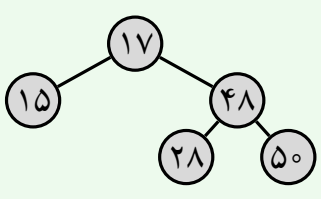
شکل ۱۱: $LR\ Rotation$



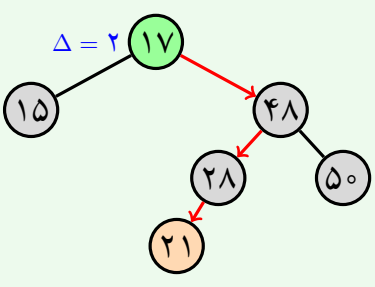
شکل ۱۲: $insert(۴۸)$



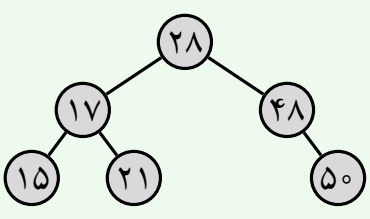
شکل ۱۳: $insert(۵۰)$



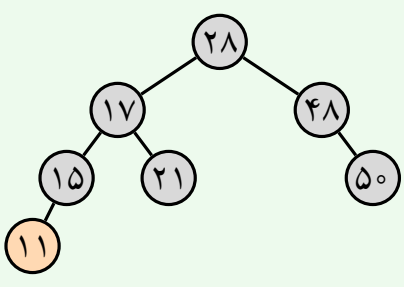
شکل ۱۴: $RR\ Rotation$



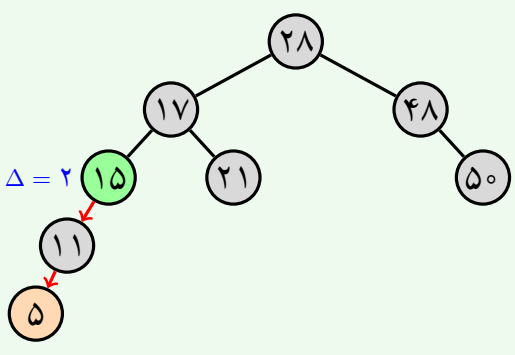
شکل ۱۵: $insert(۲۱)$



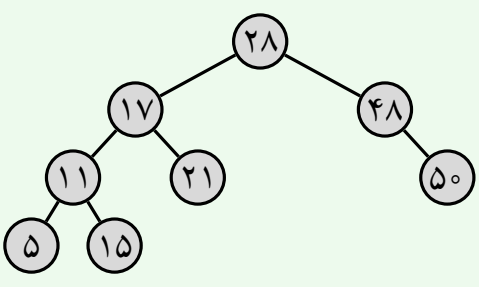
شکل ۱۶: $RL\ Rotation$



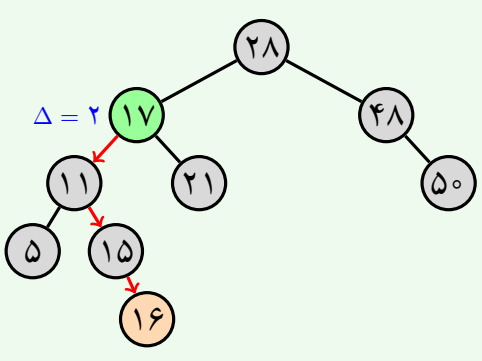
شکل ۱۷: $insert(۱۱)$



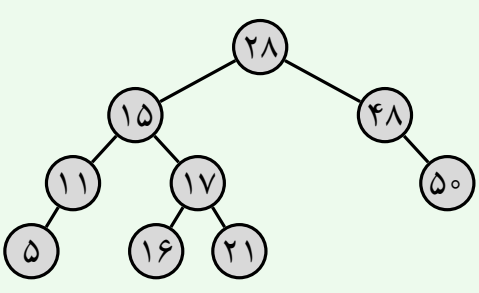
شکل ۱۸: $insert(۵)$



شکل ۱۹: $LL\ Rotation$



شکل ۲۰: $insert(۱۶)$



شکل ۲۱: $LR\ Rotation$