دانشگاه هرمزگان
دانشکده فنی و مهندسی
گروه مهندسی برق و کامپیوتر

داده‌ساختارها و الگوریتم‌ها

جلسه بیست‌ویکم: درخت AVL

دکتر شهرام گلزاری

پاییز ۱۴۰۵

اهداف جلسه

در پایان این جلسه انتظار داریم که:

- درخت AVL و مؤلفه‌ی توازن هر گره را تعریف کنیم و تشخیص دهیم یک درخت دودویی جست‌وجو ویژگی AVL دارد یا نه.
- بتوانیم ثابت کنیم ارتفاع درخت AVL با n گره همواره $O(\lg n)$ است.
- دوران‌های تکی (SLR، SRR) و دوتایی (DLR، DRR) را روی یک گره اجرا کنیم.
- حالت‌های مختلف درج در درخت AVL را تشخیص دهیم، با دوران مناسب توازن را بازیابی کنیم و توضیح دهیم چرا کل درج $O(\lg n)$ است.
- تفاوت پیچیدگی تعداد دوران‌ها در درج و حذف درخت AVL را بیان کنیم.

تعریف درخت AVL

درخت AVL چیست؟

«درخت AVL اولین درخت دودویی جست‌وجو است که مانند درخت قرمز–سیاه خود را متوازن نگه می‌دارد و ارتفاع آن همیشه لگاریتمی است. این درخت در سال ۱۹۶۲ توسط «جی.ام. اندرسن–ولسکی» و «ای.ام. لندیس» پیشنهاد شد و نام خود را از حروف اول نام این طراحان گرفت.»

تعریف ۷-۱

«درخت AVL درخت دودویی جست‌وجویی است که اختلاف ارتفاع‌های دو زیردرخت هر گره آن حداکثر یک واحد است.»

«در این تعریف ارتفاع یک زیردرخت تهی ۱−، و ارتفاع یک گره تک، هم‌چون گذشته ۰ است.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۲۹

مؤلفه‌ی توازن (Balance Factor)

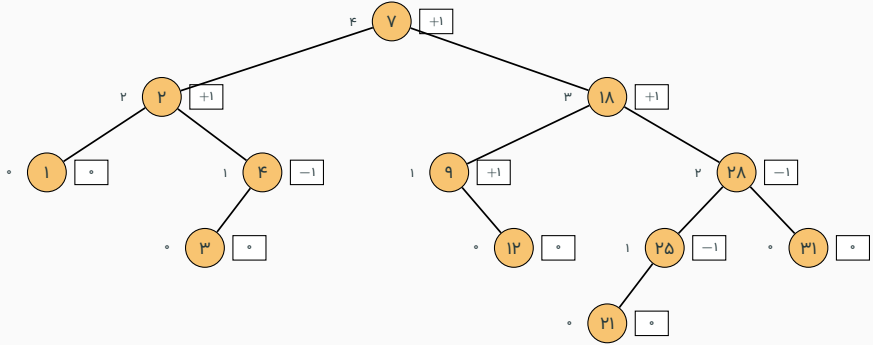
«کلید گره‌ها خاصیت د.د.ج دارند. برای پیاده‌سازی این درخت یک مؤلفه‌ی «توازن» به هر گره اضافه می‌کنیم که برابر است با اختلاف ارتفاع زیردرخت راست با زیردرخت چپ. بنابراین مؤلفه‌ی توازن مقادیر ۰، ۱+ یا ۱− را خواهد گرفت و در پیاده‌سازی به دو بیت اضافی در هر گره نیاز داریم.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۲۹

نمایش فرمولی:

$$\text{توازن}(v) \;=\; h(\text{زیردرخت راست } v) \;-\; h(\text{زیردرخت چپ } v)$$

مثالی از یک درخت AVL



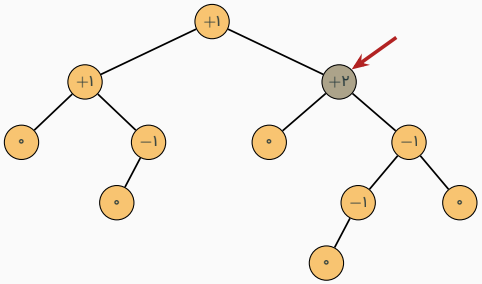
«**شکل ۷-۲۳** مثالی از یک درخت AVL. کلیدهای عناصر در داخل گره‌ها، ارتفاع هر گره در سمت چپ آن و «توازن» هر گره در سمت راست آن در داخل جعبه نشان داده شده است.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۰ (بازترسیم شکل ۷-۲۳)

مثالی از یک درخت غیر AVL

«شکل ۷-۲۴ یک درخت غیرِ AVL را نشان می‌دهد که در آن یک گره که با پیکان مشخص شده است، این خاصیت را نقض می‌کند، چون عدد توازن آن $+۲$ است. مانند این شکل و در ادامه، درخت‌های AVL را فقط با عدد توازن در داخل هر گره نشان می‌دهیم.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۲۹



بررسی گره‌ی مشخص‌شده:

- ارتفاع زیردرخت چپ: ۰
- ارتفاع زیردرخت راست: ۲
- عدد توازن: $+۲ = ۰ - ۲$

«**شکل ۷-۲۴** مثالی از یک درختِ غیرِ AVL. گره نشان‌داده شده خاصیت AVL را نقض می‌کند.»

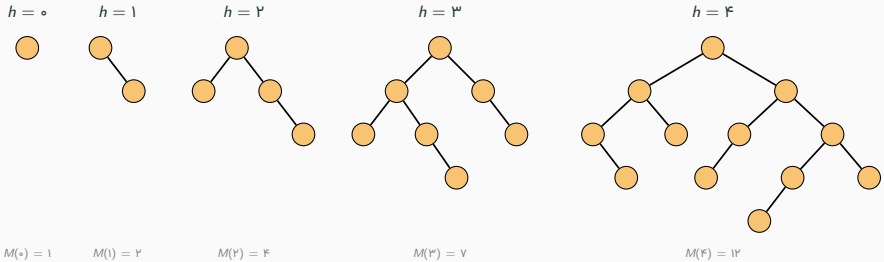
داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۰ (بازترسیم شکل ۷-۲۴)

ارتفاع درخت AVL

درخت‌های AVL خلوت (Sparse)

«برای محاسبه‌ی ارتفاع بیشینه‌ی یک درخت AVL با n گره، ابتدا $M(h)$، یعنی کمینه‌ی تعداد گره‌های یک درخت AVL با ارتفاع h را محاسبه می‌کنیم. (...) به این درخت‌ها «خلوت» می‌گوییم. بدیهی است که این درخت‌ها تک نیستند.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۲۹ و ۴۳۱



«**شکل ۷-۲۵** درخت‌های AVL «خلوت»؛ با کم‌ترین تعداد گره‌ها و با ارتفاع‌های ۰ تا ۴.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۱ (بازترسیم شکل ۷-۲۵؛ برچسب‌های $M(h)$ زیر درخت‌ها به شکل اضافه شده است)

رابطه‌ی بازگشتی $M(h)$ و دنباله‌ی فیبوناچی

«اگر ارتفاع درخت خلوت h باشد، حتماً ارتفاع یکی از زیردرخت‌های آن $h-1$ است و در آن‌صورت ارتفاع زیردرخت دوم نمی‌تواند از $h-2$ کم‌تر باشد. برای آن‌که تعداد گره‌ها کمینه شود، باید زیردرخت‌ها هم کم‌ترین ارتفاع را داشته باشند، و نیز هر یک خلوت باشد (کم‌ترین تعداد گره را داشته باشد). پس، رابطه‌ی بازگشتی زیر برقرار است:»

h	۰	۱	۲	۳	۴	۵	۶
$M(h)$	۱	۲	۴	۷	۱۲	۲۰	۳۳
$F(h)$	۱	۱	۲	۳	۵	۸	۱۳

$$M(۰) = ۱$$
$$M(۱) = ۲$$
$$M(h) = M(h-۱) + M(h-۲) + ۱, \quad \text{برای } h > ۱ \qquad (۷\text{-}۷)$$

«روشن است که $M(h) = F(h+۲) - ۱$. چنان‌چه در بخش ۳-۶ دیدیم، براساس رابطه‌ی ۳-۳۵ داریم»

$$F(h) = \frac{۱}{\sqrt{۵}} \left[\left(\frac{۱+\sqrt{۵}}{۲} \right)^h - \left(\frac{۱-\sqrt{۵}}{۲} \right)^h \right]. \qquad (۸\text{-}۷)$$

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۱

اثبات لگاریتمی بودن ارتفاع درخت AVL

«$\phi = \frac{1+\sqrt{5}}{2} \approx 1{,}618$ «نسبت طلایی» نامیده می‌شود. پس $F(r) > \frac{\phi^r}{\sqrt{5}} - 1$.»

«اگر n تعداد گره‌های درخت AVL به ارتفاع h باشد، داریم»

$$n \geq F(h+2) - 1 > \frac{\phi^{h+2}}{\sqrt{5}} - 2.$$

«بنابراین،»

$$\log_\phi(n+2) > \log_\phi\left(\frac{\phi^{h+2}}{\sqrt{5}}\right) = h + 2 - \log_\phi \sqrt{5}$$

«که از آن نتیجه می‌شود: $h < 1{,}4404 \lg(n+2) - 0{,}328$. از طرفی می‌دانیم که $h \geq \lg(n+1) - 1$ (حالت مساوی وقتی است که درخت کاملاً متوازن باشد). پس،»

$$\lg(n+1) - 1 \leq h < 1{,}44 \lg(n+2) - 0{,}328 \qquad (۹\text{-}۷)$$

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۲

دوران در درخت AVL

دوران‌های تکی و دوتایی

«برای تنظیم ارتفاع زیردرخت‌ها در اعمال درج و حذف، علاوه بر دوران راست‌گرد و چپ‌گرد بیان‌شده در بخش ۷-۳-۱، دو «دوران دوتایی» نیز تعریف می‌شوند. این دوران‌های دوتایی چپ‌گرد (به‌اختصار DLR) و راست‌گرد (به‌اختصار DRR) به‌ترتیب در شکل‌های ۷-۲۶ و ۷-۲۷ به تصویر کشیده شده‌اند و هر کدام شامل دو «دوران تکی» هستند. دوران‌های تکی نیز یا چپ‌گرد هستند یا راست‌گرد و آن‌را به‌اختصار با SLR و SRR نام‌گذاری می‌کنیم.»

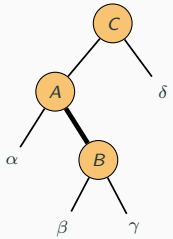
داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۲

یادآوری دوران تکی :



ترتیب میان‌ترتیب در هر دو حالت $\alpha < A < \beta < B < \gamma$ باقی می‌ماند (خاصیت BST حفظ می‌شود).

دوران دوتایی چپ‌گرد (DLR)

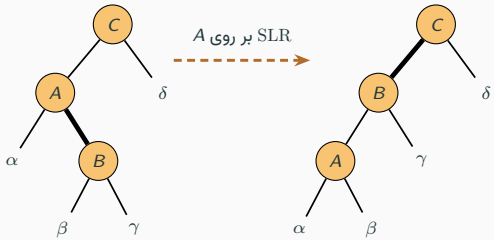


«**شکل ۷-۲۶** دوران دوتایی چپ‌گرد بر روی گره C.»

توضیح: شامل یک SLR بر روی فرزند چپ (A) و سپس یک SRR بر روی خود C.

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۶)

دوران دوتایی چپ‌گرد (DLR)

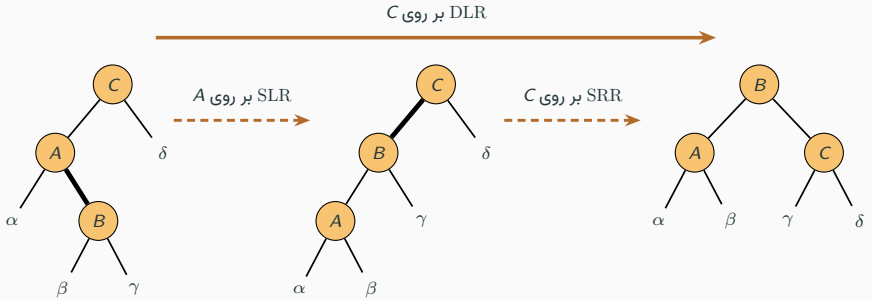


«**شکل ۷-۲۶** دوران دوتایی چپ‌گرد بر روی گره C.»

توضیح: شامل یک SLR بر روی فرزند چپ (A) و سپس یک SRR بر روی خود C.

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۶)

دوران دوتایی چپ‌گرد (DLR)

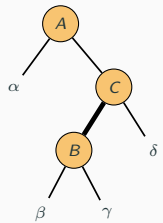


«**شکل ۷-۲۶** دوران دوتایی چپ‌گرد بر روی گره C.»

توضیح: شامل یک SLR بر روی فرزند چپ (A) و سپس یک SRR بر روی خود C.

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۶)

دوران دوتایی راست‌گرد (DRR)



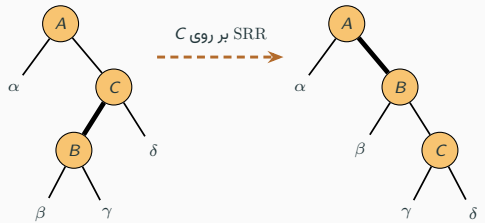
«**شکل ۷-۲۷** دوران دوتایی راست‌گرد بر روی گره A.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۷)

«از این شکل‌ها می‌توان دریافت که انجام این دوران‌ها برروی x توازن گره‌های بالاتر از x (یعنی از x تا ریشه‌ی درخت) را به‌هم نمی‌زند.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۲

دوران دوتایی راست‌گرد (DRR)



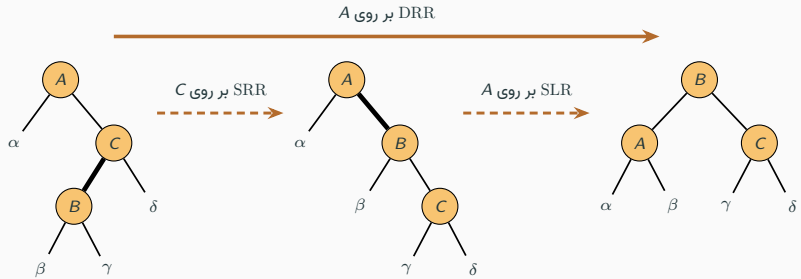
«**شکل ۷-۲۷** دوران دوتایی راست‌گرد بر روی گره A.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۷)

«از این شکل‌ها می‌توان دریافت که انجام این دوران‌ها برروی x توازن گره‌های بالاتر از x (یعنی از x تا ریشه‌ی درخت) را به‌هم نمی‌زند.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۲

دوران دوتایی راست‌گرد (DRR)



«**شکل ۷-۲۷** دوران دوتایی راست‌گرد بر روی گره A.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۳ (بازترسیم شکل ۷-۲۷)

«از این شکل‌ها می‌توان دریافت که انجام این دوران‌ها برروی x توازن گره‌های بالاتر از x (یعنی از x تا ریشه‌ی درخت) را به‌هم نمی‌زند.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۲

درج و حذف در درخت AVL

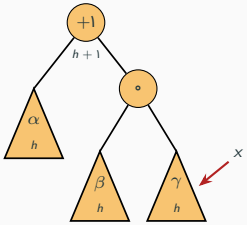
درج در درخت AVL: روند کلی

«برای درج یک عنصر x ابتدا آن‌را به‌صورت عادی در د.د.ج درج می‌کنیم و سپس با دنبال کردن اشاره‌گر «پدر»، عدد توازن گره‌های این مسیر را مورد بررسی قرار می‌دهیم و آن‌ها را اصلاح می‌کنیم. اولین گره‌ای را که عدد توازنش نادرست است، با دوران اصلاح می‌کنیم. برای این‌کار باید حالت‌های مختلفی را در نظر بگیریم:»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴

حالت اول درج: دوران تکی SLR

«**حالت اول:** وقتی است که پس از عمل درج، عدد توازن یک گره y برابر $+2$ و عدد توازن فرزند راستش $+1$ شود (شکل ۷-۲۸). این مشکل با انجام عمل SLR بر روی y حل می‌شود.»

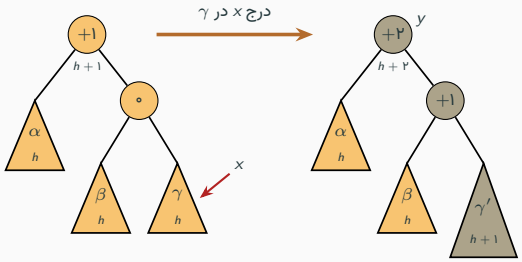


«**شکل ۷-۲۸** حالت اول درج x در درخت AVL، اگر x در زیردرخت راست فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴ (بازترسیم شکل ۷-۲۸)

حالت اول درج: دوران تکی SLR

«**حالت اول:** وقتی است که پس از عمل درج، عدد توازن یک گره y برابر $+2$ و عدد توازن فرزند راستش $+1$ شود (شکل ۷-۲۸). این مشکل با انجام عمل SLR بر روی y حل می‌شود.»

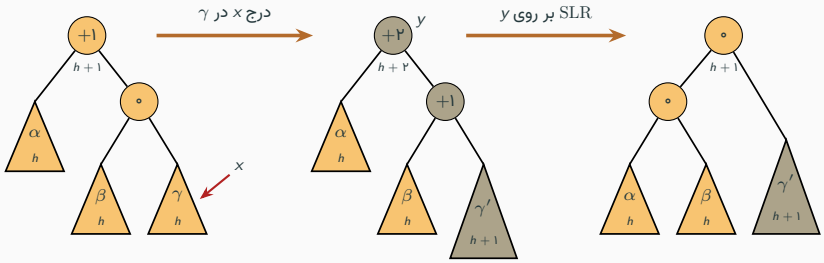


«**شکل ۷-۲۸** حالت اول درج x در درخت AVL، اگر x در زیردرخت راست فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴ (بازترسیم شکل ۷-۲۸)

حالت اول درج: دوران تکی SLR

«**حالت اول:** وقتی است که پس از عمل درج، عدد توازن یک گره y برابر $+2$ و عدد توازن فرزند راستش $+1$ شود (شکل ۷-۲۸). این مشکل با انجام عمل SLR بر روی y حل می‌شود.»

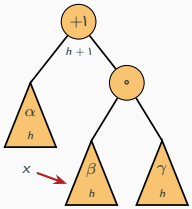


«**شکل ۷-۲۸** حالت اول درج x در درخت AVL، اگر x در زیردرخت راست فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴ (بازترسیم شکل ۷-۲۸)

حالت دوم درج (الف): درج در زیردرخت چپ z

«**حالت دوم:** وقتی است که عددهای توازن یک گره y برابر $+2$ و فرزند راستش به‌نام w (با فرزند چپ z) برابر -1 شود (شکل ۷-۲۹). این خود دو حالت دارد: x در زیردرخت چپ (حالت (الف)) یا راست (...) z درج شده باشد. این دو مشکل با انجام یک دوران دوتایی بر روی y حل می‌شود.»

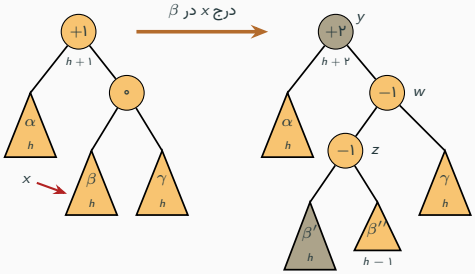


«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه‌های ۴۳۴ و ۴۳۵ (بازترسیم شکل ۷-۲۹ الف)

حالت دوم درج (الف): درج در زیردرخت چپ z

«**حالت دوم:** وقتی است که عددهای توازن یک گره y برابر $+۲$ و فرزند راستش به‌نام w (با فرزند چپ z) برابر -۱ شود (شکل ۷-۲۹). این خود دو حالت دارد: x در زیردرخت چپ (حالت (الف)) یا راست (...) z درج شده باشد. این دو مشکل با انجام یک دوران دوتایی بر روی y حل می‌شود.»

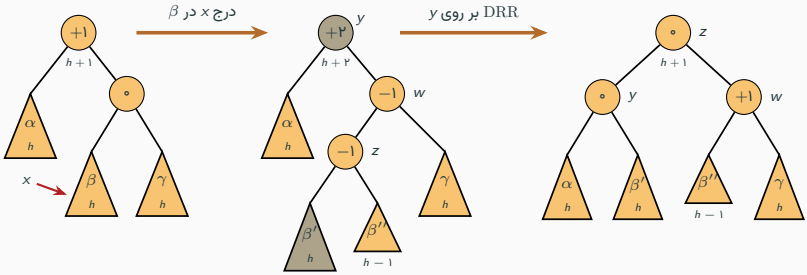


«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه‌های ۴۳۴ و ۴۳۵ (بازترسیم شکل ۷-۲۹ الف)

حالت دوم درج (الف): درج در زیردرخت چپ z

«**حالت دوم:** وقتی است که عددهای توازن یک گره y برابر $+2$ و فرزند راستش به‌نام w (با فرزند چپ z) برابر -1 شود (شکل ۷-۲۹). این خود دو حالت دارد: x در زیردرخت چپ (حالت (الف)) یا راست (...) z درج شده باشد. این دو مشکل با انجام یک دوران دوتایی بر روی y حل می‌شود.»

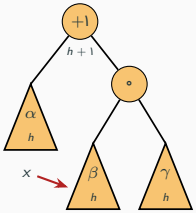


«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه‌های ۴۳۴ و ۴۳۵ (بازترسیم شکل ۷-۲۹ الف)

حالت دوم درج (ب): درج در زیردرخت راست z

توضیح حالت (ب): همانند حالت قبل، توازن y برابر $+2$ و توازن فرزند راستش w برابر -1 است؛ این‌بار x در زیردرخت راست z (β'') درج شده و باز هم با DRR بر روی y اصلاح می‌شود.

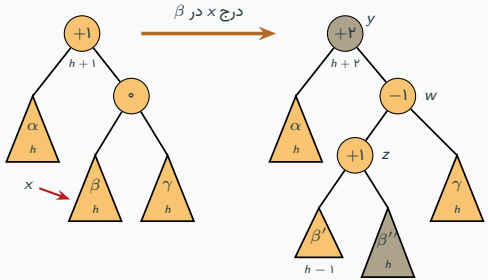


«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۵ (بازترسیم شکل ۷-۲۹ ب)

حالت دوم درج (ب): درج در زیردرخت راست z

توضیح حالت (ب): همانند حالت قبل، توازن y برابر $+۲$ و توازن فرزند راستش w برابر -۱ است؛ این‌بار x در زیردرخت راست z (β'') درج شده و باز هم با DRR بر روی y اصلاح می‌شود.

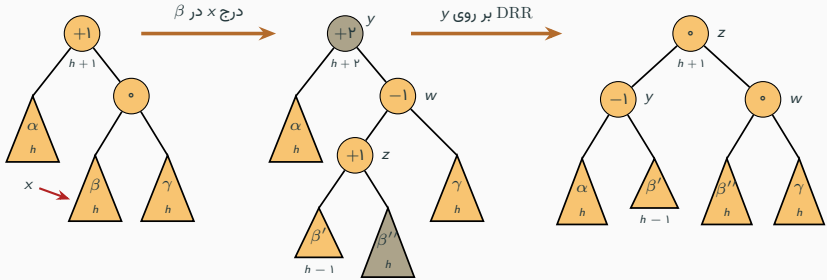


«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۵ (بازترسیم شکل ۷-۲۹ ب)

حالت دوم درج (ب): درج در زیردرخت راست z

توضیح حالت (ب): همانند حالت قبل، توازن y برابر $+2$ و توازن فرزند راستش w برابر -1 است؛ این‌بار x در زیردرخت راست z (β'') درج شده و باز هم با DRR بر روی y اصلاح می‌شود.



«**شکل ۷-۲۹** حالت دوم درج x در درخت AVL، اگر x در زیردرخت چپ فرزند راست درج شود و توازن به‌هم بخورد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۵ (بازترسیم شکل ۷-۲۹ ب)

تحلیل هزینه‌ی درج در درخت AVL

«حالت‌های قرینه‌ی دیگر نیز مشابه هستند. توجه کنید که چون پس از یک یا دو دوران، ارتفاع گره‌ای که عدد توازنش به‌هم خورده است، به همان مقدار قبل از درج برمی‌گردد، اعداد توازن گره‌های بالاتر تا ریشه نیز تغییری نمی‌کند. بنابراین هزینه‌ی عمل اصلاح توازن درخت پس از درج، $O(1)$ است. در نتیجه، کل عمل درج $O(\lg n)$ است.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴

جمع‌بندی هزینه‌ی درج

- یافتن محل درج و پیمایش مسیر: $O(\lg n)$
- اصلاح توازن با دوران: $O(1)$
- هزینه‌ی کل عمل درج: $O(\lg n)$

نتیجه‌ی تجربی

«برحسب داده‌های تجربی، حدود ۵۳٪ از اعمال درج در درخت AVL، نیازی به انجام دوران ندارد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۴

حذف از درخت AVL

«عمل حذف در درخت AVL به سادگی درج که گفتیم نیست و حالت‌های مختلف زیادی دارد و ممکن است به تعداد $O(\lg n)$ دوران نیاز داشته باشد.»

داده‌ساختارها و مبانی الگوریتم‌ها، محمد قدسی، صفحه ۴۳۵

منابع

منابع

- محمد قدسی، داده‌ساختارها و مبانی الگوریتم‌ها، انتشارات فاطمی، چاپ ۱۳۹۷.