Tree
(ادامه)

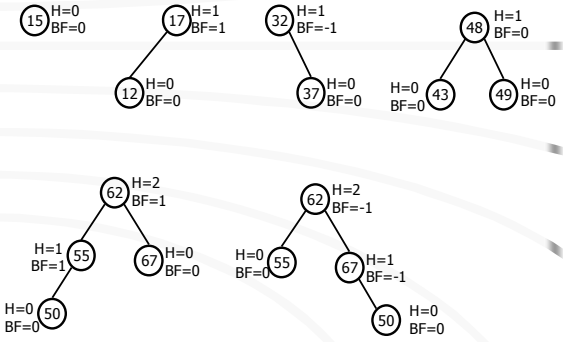
انگیزه

- درخت دودویی جستجو (BST) در حالتی که کامل باشد ساختار مناسبی برای عملیات دیکشنری است ((O(lgn)
- در بدترین حالت، درخت دودویی جستجو به یک لیست پیوندی تبدیل می شود ((O(n)
- روشهایی وجود دارد که درخت دودویی جستجو (نزدیک به) دودویی کامل نگهداری شود
  - AVL (Adelson, Velskii, Landis)
  - درخت ۲-۳ (2-3- Tree)
  - درخت ۲-۳-۴ (2-3-4- Tree)
  - درخت قرمز سیاه (Red Black Tree)
  - . . .

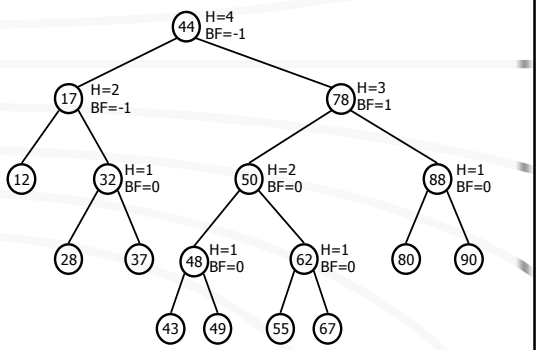
AVL tree

- طبق تعریف: ضریب تعادل (BF:Balance Factor) برای هر گره T در درخت دودویی برابر است با $BF=h_L-h_R$ که در آن h_L و h_R به ترتیب ارتفاع زیردرختان چپ و راست T است.
- در AVL tree ، BF برای هر گره درخت یکی از مقادیر 0،1، یا 1- است.

مثال درخت AVL



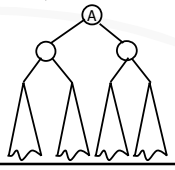
مثال درخت AVL



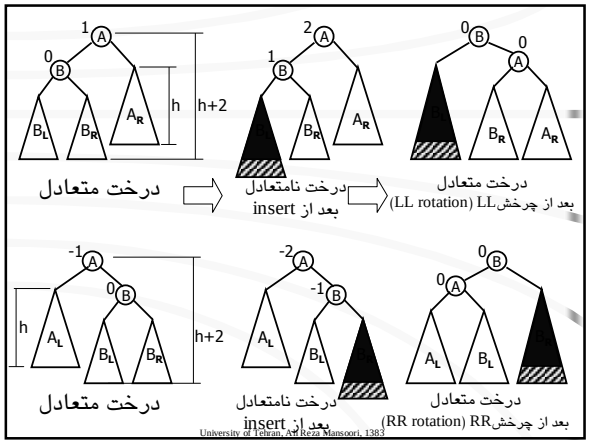
انواع چرخش در AVL

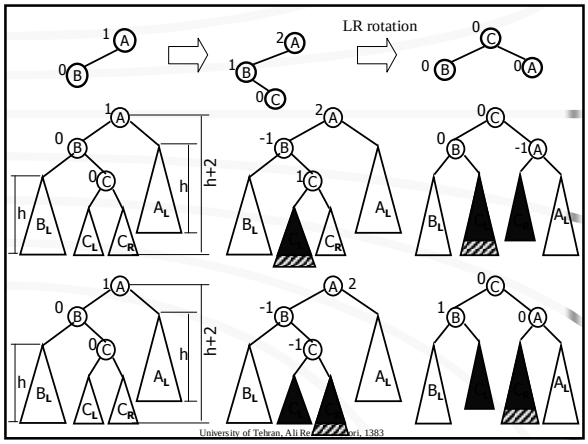
با اضافه شدن گره ای به درخت AVL ممکن است تعادل درخت از بین برود (گره ای با $BF=\pm 2$ به وجود آید) برای متعادل ساختن درخت پس از اضافه شدن گره ای چرخش (rotation) انجام می شود.

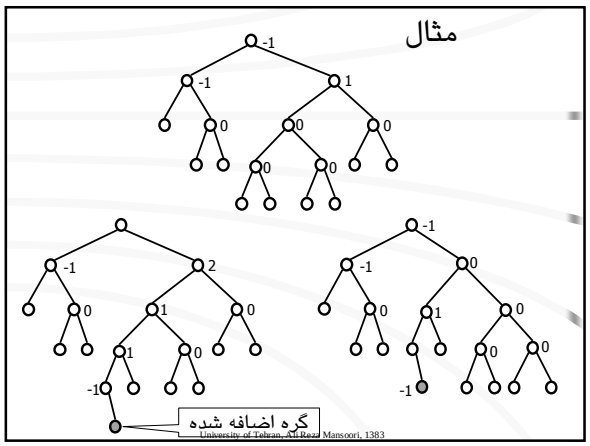
انواع چرخش (rotation) در درخت AVL بر اساس محل اضافه شده گره Y نسبت به گره A است (گره A نزدیکترین جد به گره اضافه شده Y است که پس از اضافه شدن Y، BF برای آن گره ± 2 می شود)

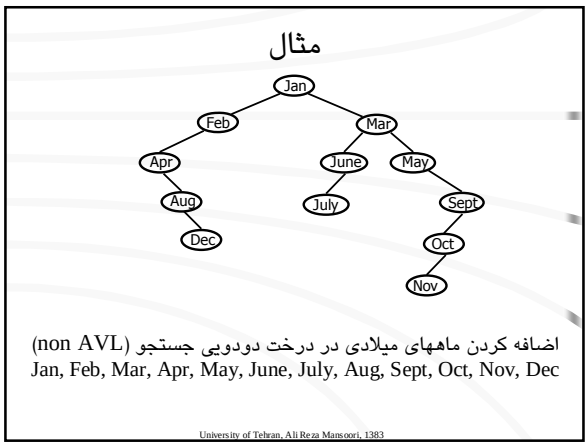


- Left subtree of Left subtree : LL
- Right subtree of Right subtree : RR
- Left subtree of Right subtree :RL
- Right subtree of Left subtree :LR



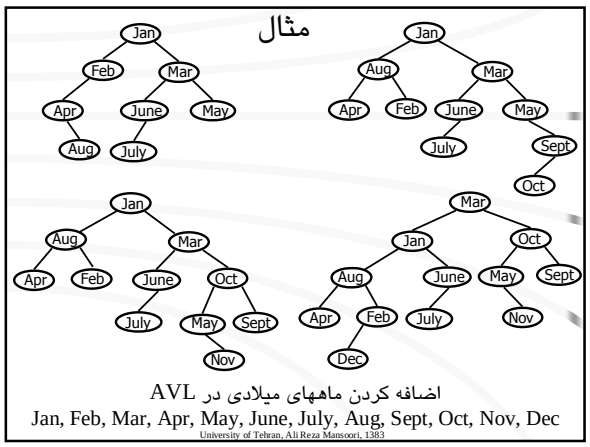






اضافه کردن ماههای میلادی در درخت دودویی جستجو (non AVL)
Jan, Feb, Mar, Apr, May, June, July, Aug, Sept, Oct, Nov, Dec

University of Tehran, Ali Reza Mansoori, 1383



نکاتی برای پیاده سازی AVL

- در الگوریتم insert باید گره A (که BF ، 2± می شود) و پدر A را در اختیار داشت
- اگر گره A در طی insert ایجاد شود، یکی از گره هایی است که BF برای آن 1± بوده است
- برای دستیابی به گره A در الگوریتم delete می توان از پیوند دوطرفه در درخت استفاده کرد.

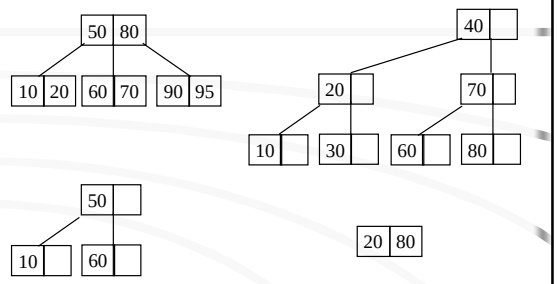
University of Tehran, Ali Reza Mansoori, 1383

درخت ۳-۲ (2-3 Tree)

درخت ۳-۲ درختی است که یا تهی است یا شرایط زیر را دارد:

- هر گره یا 2-node است یا 3-node (در 2-node یک عنصر و در 3-node دو عنصر قرار می گیرد)
- در 2-node: عنصر Lkey و دو زیردرخت LChild و MChild که کلیدهای زیردرخت LChild از Lkey کوچکتر و کلیدهای زیردرخت MChild از LKey بزرگتر هستند.
- در 3-node: دو عنصر Lkey و Rkey و سه زیردرخت LChild، MChild، و RChild با شرایط زیر:
  - Lkey < RKey +
  - + کلیدهای زیردرخت LChild از LKey کوچکتر هستند
  - + کلیدهای زیردرخت MChild از LKey بزرگتر و از RKey کوچکتر هستند
  - + کلیدهای زیردرخت RChild از Rkey بزرگتر هستند
- همه برگها در یک سطح قرار دارند

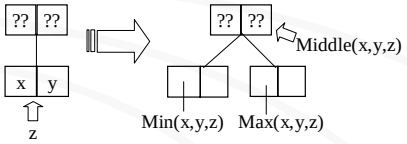
مثال 2-3 Tree



درج عنصر در 2-3 Tree

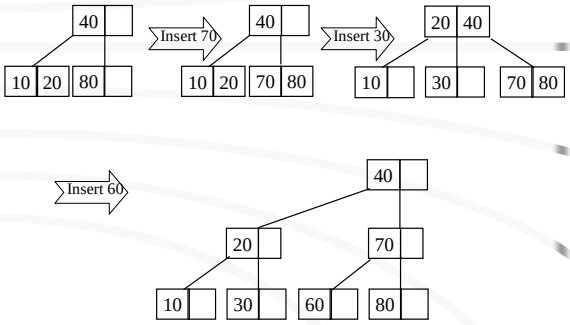
– عنصر جدید به یک برگ اضافه می شود. این برگ با جستجوی عنصر جدید پیدا می شود.

- اگر برگ، یک 2-node باشد، به 3-node تبدیل می شود و عنصر جدید به آن اضافه می شود (با رعایت ترتیب نسبی عناصر 3-node)
- اگر برگ، یک 3-node باشد، تقسیم (split) اتفاق می افتد و یک عنصر (عنصر با مقدار کلید میانی بین سه کلید) به گره پدر اضافه می شود:



– اگر ریشه درخت تقسیم (split) شود، یک ریشه جدید اضافه می شود و ارتفاع درخت یک واحد اضافه می شود.

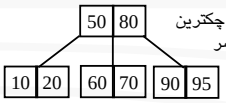
مثال درج عنصر در 2-3 Tree



ارتفاع درخت یک واحد اضافه می شود

حذف عنصر در 2-3 Tree

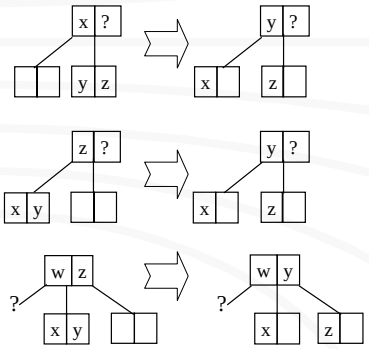
- اگر عنصری که باید حذف شود در برگ نیست با یک عنصر در یکی از برگها تعویض می شود



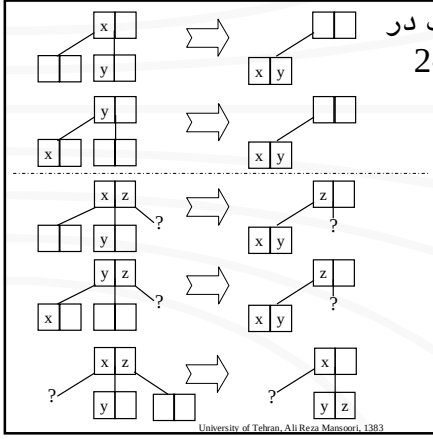
مثلا در درخت مقابل برای حذف 50، با 60 (کوچکترین عنصر زیردرخت راست) یا 20 (بزرگترین عنصر زیردرخت چپ) تعویض می شود

- اگر عنصر از یک 3-node حذف شود، 3-node به 2-node تبدیل می شود.
- اگر عنصر از یک 2-node حذف می شود، چرخش (rotation) یا ترکیب (combine) انجام می شود.

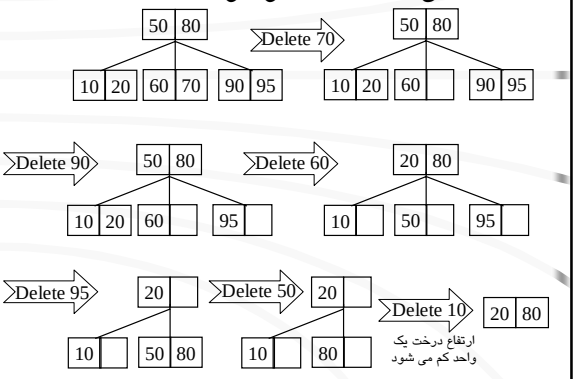
انواع چرخش در 2-3 Tree



انواع ترکیب در 2-3 Tree

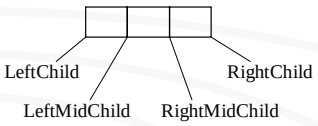


مثال حذف عنصر در 2-3 Tree

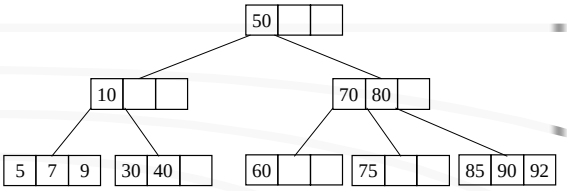


درخت ۴-۳-۲ 2-3-4 Tree

درخت ۴-۳-۲ گسترشی از درخت ۳-۲ است که در آن علاوه بر 2-node و 3-node ، 4-node نیز وجود دارد.
در 4-node سه کلید و چهار زیردرخت وجود دارد:



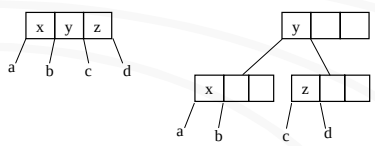
مثال 2-3-4 Tree



درج عنصر در 2-3-4 Tree

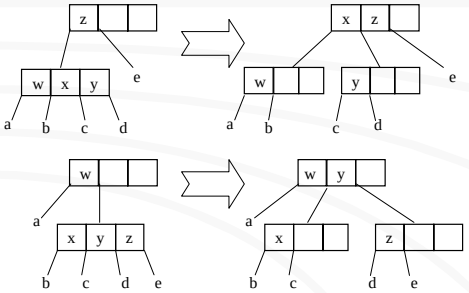
در مسیری که از ریشه تا برگ محل درج شدن عنصر طی می شود، اگر به 4-node برخورد شود، تقسیم (split) انجام می شود
در تقسیم هر 4-node حالتهای مختلفی ممکن است :

- حالت اول: 4-node، ریشه درخت باشد:



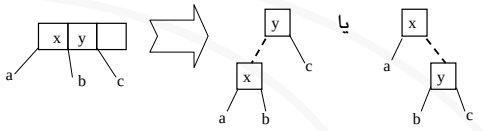
درج عنصر در 2-3-4 Tree

- حالت دوم: پدر 4-node یک 2-node باشد:



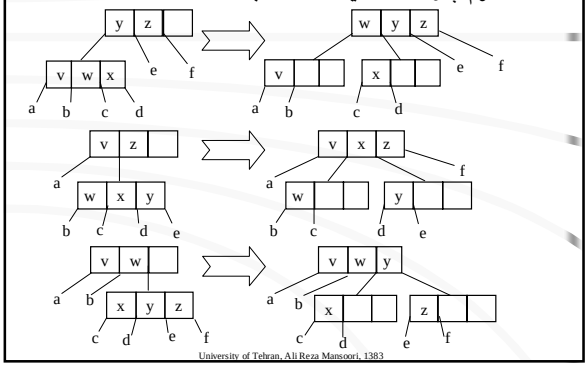
درخت قرمز سیاه (Reb Black Tree)

- درخت قرمز سیاه نمایش دودویی درخت ۴-۳-۲ است
- اشاره گر به فرزندان در هر گره یا قرمز است یا سیاه
- اگر اشاره گر به فرزند، در درخت ۴-۳-۲ وجود دارد، در درخت قرمز سیاه، سیاه است و در غیر این صورت، یک اشاره گر قرمز است
- به هر گره، یکی از رنگهای قرمز یا سیاه نسبت داده می شود که همرنگ اشاره گر از پدر است
- طبق تعریف، رنگ گره ریشه سیاه است



درج عنصر در 2-3-4 Tree

– حالت سوم: پدر 4-node یک 3-node باشد:

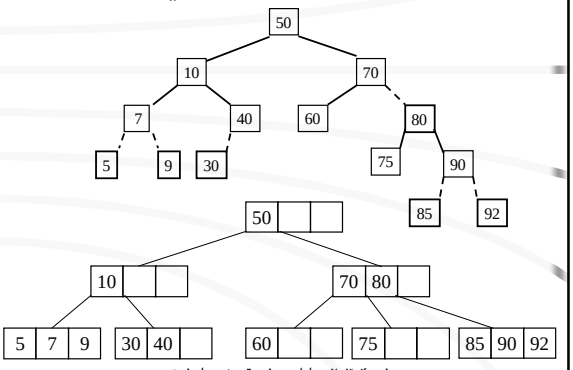


تعریف درخت قرمز سیاه

درخت قرمز سیاه یک درخت دودویی جستجو است که خواص زیر را دارد:

- هر گره یا قرمز است یا سیاه
- هر برگی سیاه است (NIL)
- اگر گره ای قرمز است، فرزندانش سیاه است
- هر مسیر ساده ای از هر گره به یکی از برگهای زیردرخت آن گره شامل تعداد مساوی گره های سیاه است (black height)

مثال درخت قرمز سیاه

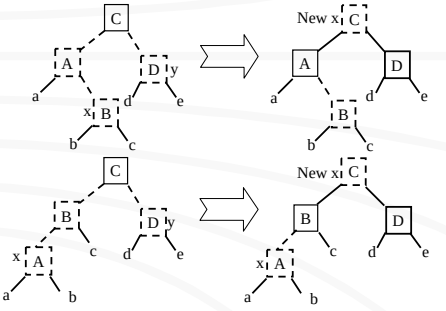


درخت ۴-۳-۲ معادل درخت قرمز سیاه فوق

درج در درخت قرمز سیاه

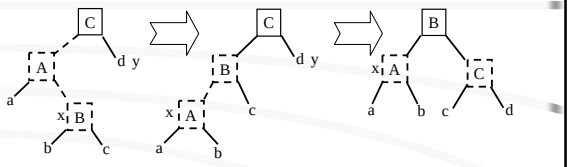
- مشابه درج در BST ، عنصر جدید x به درخت اضافه می شود
- گره اضافه شده، قرمز رنگ می شود
- تا برقرار شدن شرایط درخت قرمز سیاه، با عملیات چرخش و تغییر رنگ برای حالتهای مختلف عمل درج انجام می شود

تغییر رنگ در درخت قرمز سیاه



حالتی که y برادر چپ پدر x باشد، مشابه حالات فوق است

چرخش در درخت قرمز سیاه



حالتی که y برادر چپ پدر x باشد، مشابه حالات فوق است