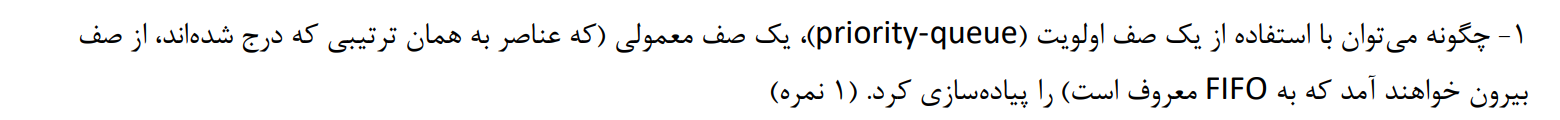
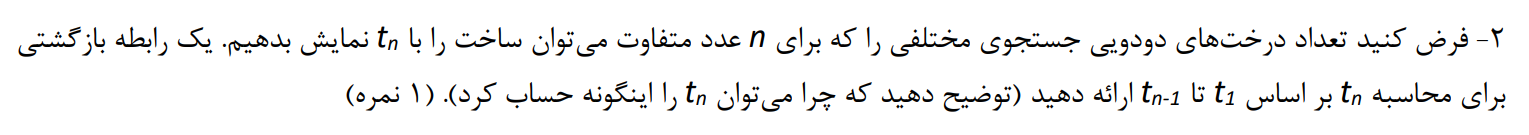
به نام خدا

تمرین شماره ۳ درس ساختمان داده‌ها و الگوریتم‌ها

چمران معینی : ۹۹۳۱۰۵۳





اگر فقط یک عنصر داشته باشیم، مشخص است که فقط یک راه برای تبدیل آن به درخت دودویی جست و جو داریم،‌ یعنی:

اگر دو عنصر داشته باشیم، دو انتخاب برای ریشه داریم. بعد از انتخابِ ریشه هم t1 حالت برای چینش بقیه‌ی عناصر، یعنی:

در حالتی هم که سه عنصر داشته باشیم، سه انتخاب برای ریشه داریم. اگر بزرگ‌ترین یا کوچک‌ترین عنصر ریشه‌مان باشد هر دو عنصرِ دیگر در یک طرف قرار می‌گیرند، یعنی t2 حالت برای چینشِ بقیه‌ی اعضا داریم. اگر هم عوض میانی را انتخاب کرده باشیم t1 \* t1 حالت برای چینشِ بقیه‌ی اعضا داریم، پس:

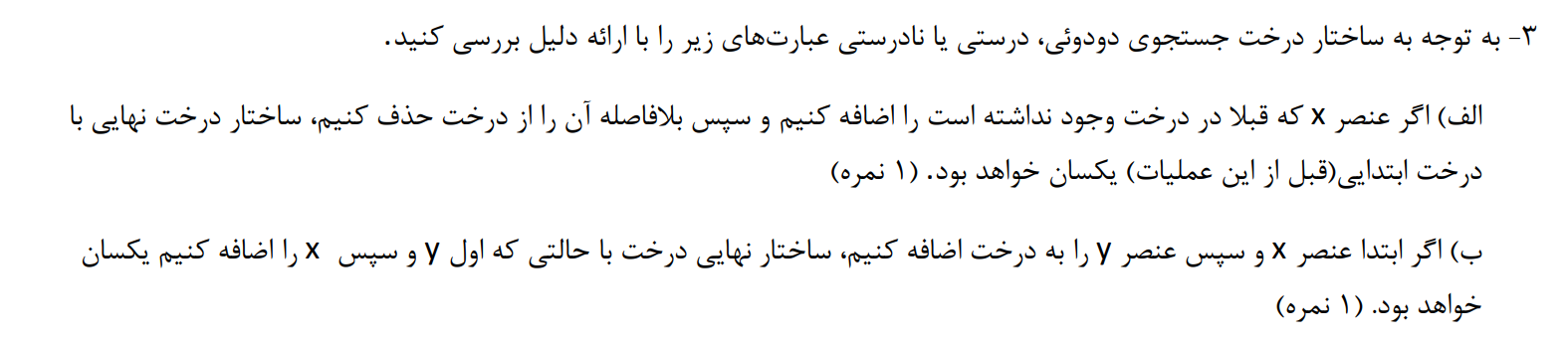
در حالتی که چهار عنصر داریم، چهار انتخاب برای ریشه داریم. در حالتی که بزرگ‌ترین یا کوچک‌ترین عضو را انتخاب کرده باشیم، هر سه عضو در یک طرف قرار می‌گیرند و t3 حالت برای چینش‌شان خواهیم داشت. در حالتی که عضو دوم یا سوم را انتخاب کرده باشیم هم t1 \* t2 حالت برای چینشِ بقیه‌ی اعضا داریم، پس:

وقتی پنج عنصر داشته باشیم هم، پنج انتخاب برای ریشه داریم. اگر بزرگ‌ترین یا کوچک‌ترین عضو باشد t4 حالت برای چینشِ بقیه‌ی عناصر داریم. اگر عضو دوم یا چهارم باشد t1 \* t3 حالت برای چینشِ بقیه‌ی اعضا داریم. اگر عضو سوم ریشه بشود هم t2 \* t2 حالت برای چینشِ بقیه‌ی اعضا داریم، پس:

به همین ترتیب و با همین منطق خواهیم داشت:

پس tn را به این شکل می‌نویسیم:

می‌توان این رابطه را به این شکل توضیح داد که از اولین جمله، مربوط به حالتی‌ست که بزرگ‌ترین عنصر را به عنوان ریشه انتخاب کرده باشیم، در این حالت تمام n-1 در یک سمت قرار می‌گیرند و می‌توان آن‌ها را به t(n-1) حالت چید. دومین جمله، مربوط به زمانی‌ست که دومین عنصر بزرگ را به عنوان ریشه انتخاب کنیم، در این حالت یک عنصر در یک سمت قرار می‌گیرد که t1 چینش دارد، بقیه‌ی اعضا هم در سمت دیگر قرار می‌گیرند که t(n-2) چینش خواهند داشت، الی آخر.



الف) می‌دانیم که عملیاتِ insert در یک ددج، تغییری در جایگاهِ قبلیِ عناصر ایجاد نمی‌کند. تنها کاری که می‌کنیم این است که می‌گردیم و خانه‌ی نال مناسب را پیدا می‌کنیم و عنصرِ جدید را به جای آن می‌گذریم.

اما می‌دانیم که عملیاتِ حذف می‌تواند با تغییراتی در عناصرِ دیگر هم همراه باشد، اما در چه صورت؟ در صورتی که عنصرمان فرزندانی داشته باشد، اما می‌دانیم که عنصری که تازه اضافه شده، به عنوان یک برگ اضافه می‌شود و فرزندی ندارد و با توجه به این که بلافاصله بعد از اضافه کردن اقدام به حذف شده، در نتیجه فقط یک لحظه عنصری به عنوان فرزندِ یکی از بزرگ‌ها اضافه می‌شود و سپس دوباره حذف می‌شود و درخت دقیقا مانند قبل خواهد بود، پس این گزاره صحیح است.

ب) اشتباه است. برای نقض آن، یک مثال نقض کافی‌ست. فرض کنید ددج‌ای داشته باشیم با ریشه‌ی ۱۰، اگر اول ۹ را اضافه کنیم و بعد ۸ را، ۹ فرزندِ ریشه می‌شود و ۸ هم فرزندِ ۹ می‌شود، اما در صورتی که اول ۸ را اضافه کرده باشیم، ۸ فرزندِ ریشه می‌شود و ۹ در سمت راستِ ۸ قرار می‌گیرد.

