یک الگوریتم خطی برای مسئله انتخاب

طراحی الگوریتم
دانشکده ریاضی. دانشگاه صنعتی خواجه نصیرالدین طوسی. پاییز ۱۴۰۲

۱ مسئله انتخاب Selection

فرض کنید A مجموعه‌ای از عناصر باشد که قابل مقایسه هستند. لیست $Sorted(A)$ شامل عناصر A است که از کوچک به بزرگ مرتب شده‌اند. برای $x \in A$، رتبه x یا $rank(x)$ در A برابر با اندیس x در لیست $Sorted(A)$ می‌باشد. در مسئله انتخاب با داشتن مجموعه عناصر A و عدد i می‌خواهیم عنصر با رتبه i را در A پیدا کنیم. روشن است که عنصر با رتبه i را می‌توان با مرتب سازی A و محاسبه $Sorted(A)$ در زمان $\Theta(n \log n)$ مثلا با استفاده از مرتب سازی ادغامی بدست آورد. اینجا می‌خواهیم نشان دهیم الگوریتمی وجود دارد که عنصر با رتبه i را در زمان $\Theta(n)$ بدست می‌آورد.

۲ توصیف الگوریتم SELECT

فرض کنید لیست ورودی A شامل n عنصر متمایز است که در اندیسهای 1 تا n ذخیره شده اند.

۱. لیست ورودی که به گروه های 5 عنصری تقسیم کن. یکی از گروهها ممکن است کمتر از 5عنصر داشته باشد.

۲. میانه هر گروه را پیدا کن.

۳. فرض کنید $m_1, \cdots, m_k$ میانه‌های بدست آمده باشند. میانه مجموعه $\{m_1, \cdots, m_k\}$ را بصورت بازگشتی، با استفاده از الگوریتم SELECT پیدا کن. فرض کن میانه محاسبه شده x باشد.

۴. عناصر A را به گروه L و R تقسیم کن. گروه L آنهایی هستند که از x کمتر هستند و گروه R آنهایی هستند که از x بزرگتر هستند. دقت کنید اینجا $rank(x)$ مشخص است و دقیقا برابر با $|L| + 1$ است.

۵. سه حالت وجود دارد.

(آ) اگر $rank(x) = i$ آنگاه x لزوما عنصر با رتبه i است که دنبالش می‌گشتیم (پایان).

(ب) اگر $rank(x) > i$ پس عنصر با رتبه i در لیست L قرار دارد. در این حالت الگوریتم SELECT را برای عناصر L و رتبه i بصورت بازگشتی فراخوانی می‌کنیم.

(ج) اگر $rank(x) < i$ پس عنصر با رتبه i در لیست R قرار گرفته است. در این حالت الگوریتم SELECT را برای عناصر R و رتبه $i - rank(x)$ فراخوانی می‌کنیم.

۳ تحلیل زمان اجرای SELECT

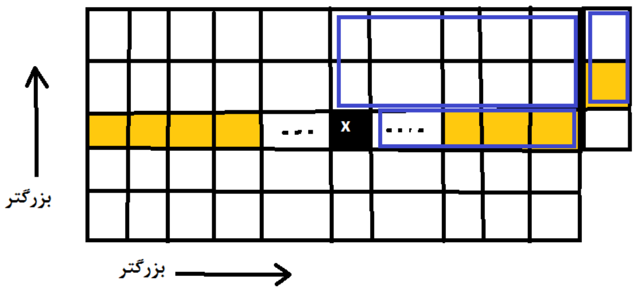
اگر $T(n)$ زمان اجرای الگوریتم SELECT باشد یک رابطه بازگشتی برای $T(n)$ می‌توان نوشت. می‌توان فرض کرد که اگر اندازه لیست ورودی از حد آستانه‌ای کمتر بود مثلا اگر $n \leq b$ آنگاه عنصر با رتبه i را با استفاده از مرتب سازی بدست می آوریم. داریم

$$T(n) = \begin{cases} T(\lceil n/5 \rceil) + T(\max\{|L|, |R|\}) + O(n) & n > b \\ O(1) & n \leq b \end{cases}$$

دقت کنید اینجا b یک عدد ثابت است که مقدارش بعدا مشخص می‌شود. همچنین همه کارهایی که الگوریتم (غیر از فراخوانی‌های بازگشتی) انجام می‌دهد، یعنی مراحل ۱ و ۲ و ۴ در زمان $O(n)$ قابل پیاده‌سازی است.

لم. $\max\{|L|, |R|\} \leq \frac{7}{10}n + 6$

اثبات. نشان می‌دهیم تعداد عناصر کمتر x حداکثر $\frac{7}{10}n + 6$ است. یعنی $|L| \leq \frac{7}{10}n + 6$.
با قطعیت می‌توان گفت تعدادی از عناصر بزرگتر از x هستند. به شکل زیر توجه کنید.



گروه های پنج تایی را با در نظر گرفتن ترتیب میانه‌ها از کوچک به بزرگ کنار هم قرار داده‌ایم. خانه‌های نارنجی میانه‌ها هستند. خانه‌هایی که با رنگ آبی مشخص شده‌اند، عناصر بزرگتر از x هستند. توجه کنید تقریبا نصف گروهها سه تا از عنصرهایشان از x بزرگتر است. پس می‌توان گفت

$$|R| \geq 3(\lceil \frac{1}{2} \lceil n/5 \rceil \rceil - 2) \geq \frac{3n}{10} - 6$$

توجه کنید عدد 2 را از نصف گروهها $\lceil \frac{1}{2} \lceil n/5 \rceil \rceil$ کم کرده‌ایم چون یکی از گروهها ممکن است 5 عنصر نداشته باشد و همچنین گروهی که شامل x است، فقط دو عنصرش از x بزرگتر است. در هر صورت چون $|R| + |L| < n$ از رابطه بالا نتیجه می‌شود که

$$|L| < \frac{7n}{10} + 6$$

مشابه همین توجیهات را می‌توان استفاده کرد و نشان داد که

$$|R| < \frac{7n}{10} + 6$$

$\square$

پس رابطه بازگشتی را می‌توان بصورت زیر خلاصه کنیم. اینجا a یک عدد ثابت است.

$$T(n) \leq \begin{cases} T(\lceil n/5 \rceil) + T(\frac{7n}{10} + 6) + an & n > b \\ O(1) & n \leq b \end{cases}$$

با استفاده از جدس و استقرا می‌توان نشان داد که عدد ثابت c وجود که برای $n \geq b$ همواره $T(n) \leq cn$. این یعنی اینکه $T(n) = \Theta(n)$.

$$\begin{aligned} T(n) &\leq T(\lceil n/5 \rceil) + T(\frac{7n}{10} + 6) + an \\ &\leq c(\lceil n/5 \rceil) + c(\frac{7n}{10} + 6) + an \\ &\leq \frac{9cn}{10} + 7c + an \\ &\leq cn + (-\frac{cn}{10} + 7c + an) \end{aligned}$$

این مقدار کمتر از cn اگر

$$-\frac{cn}{10} + 7c + an \leq 0$$

که این معادل $c \geq 10a\frac{n}{n-70}$ است اگر $n > 70$ باشد. در واقع اگر $n \geq 140$ و $c \geq 20a$ باشد آنگاه نامساوی بالا برقرار خواهد بود. پس کافی است قرار دهیم $b = 140$. در اینصورت همواره $T(n) = \Theta(n)$.