

ساختمان داده‌ها و الگوریتم‌ها

نیم‌سال دوم ۹۷-۹۸

گردآورندگان: علیرضا اکبری، پدram خورسندی



دانشکده‌ی مهندسی کامپیوتر

زمان: ۳۰ دقیقه

پاسخ کوئیز سری سوم

۲ اردیبهشت

مسئله‌ی ۱. نوآوری و برازندگی؟

در الگوریتم مرتب‌سازی سریع، طی یک عملیات محیرالقول، در مرحله Partition دو محور (pivot) انتخاب کرده‌ایم. برای این نوع الگوریتم مرتب‌سازی سریع، تحلیل پیچیدگی زمانی Best Case و Worst Case انجام دهید. آیا این تغییر بهبودی نسبت به حالت یک محوره ایجاد می‌کند؟ فرض کنید که در هر مرحله محورها را از ابتدا و انتهای آرایه انتخاب می‌کنیم.

حل.

تحلیل Worst Case: در بدترین حالت، اگر آرایه به صورت پیش‌فرض مرتب‌شده باشد، آنگاه partition خوبی نداریم چرا که عناصر اول و آخر را به عنوان محور در نظر گرفته بودیم و چون آرایه مرتب است، محورها در نهایت نیز در جای خود باقی می‌مانند و یک زیرمسئله $(n-2)$ بوجود می‌آید. در واقع رابطه بازگشتی آن به صورت زیر می‌شود:

$$T(n) = T(n-2) + \theta(n) \rightarrow T(n) \in \theta(n^2)$$

تحلیل Best Case: بهترین حالت زمانی اتفاق می‌افتد که انتخاب ۲ محور باعث بوجود آمدن ۳ بخش تقریباً مساوی (از لحاظ اندازه) شود. یک بخش کوچکتر از محور اول. یک بخش اعداد میان محور اول و محور دوم. و بخش آخر بخش بزرگتر از محور دوم. رابطه بازگشتی آن به صورت زیر می‌شود:

$$T(n) = 3T\left(\frac{n}{3}\right) + \theta(n)$$

که با تحلیل آن متوجه می‌شویم که

$$T(n) \in \theta(n \log_3 n)$$

که می‌دانیم از لحاظ پیچیدگی تفاوتی با $n \lg n$ ندارد. پس از لحاظ پیچیدگی تفاوتی میان حالات ۲ محوره و تک‌محوره وجود ندارد اما با توجه به آنکه $\lg n > \log_3 n$ می‌توان گفت که در عمل الگوریتم ۲ محوره باعث بهبودی کمی می‌شود.

▷

مسئله‌ی ۲. چندمین کوچکترین

آرایه نامرتب A به شما داده شده است. می‌خواهیم kامین کوچکترین عنصر آرایه (عنصر kام در

آرایه مرتب‌شده) را پیدا کنیم. الگوریتمی از مرتبه $O(k + (n - k) \times \log k)$ برای این منظور ارائه دهید.

حل. عنصر ابتدای آرایه را در یک هرم بیشینه درج می‌کنیم. این عمل $O(k)$ هزینه خواهد داشت. در ادامه سایر عناصر آرایه را با ریشه هرم مقایسه می‌کنیم. در صورت بیشتر بودن عنصر مورد بررسی از ریشه عنصر را کنار گذاشته و عنصر بعدی آرایه را بررسی می‌کنیم. در غیر این صورت، ریشه هرم را حذف کرده و عنصر مورد بررسی را در هرم درج می‌کنیم. این عملیات $O(\log k)$ هزینه دارد. در نتیجه کل رویه بالا پیچیدگی از $O(k + (n - k) \log k)$ خواهد داشت. \triangleright

مسئله ۳. d-ary heap

d-ary heap داده ساختاری مشابه با هرم عادی است. با این تفاوت که هر راس غیر از برگ آن d فرزند دارد. پیاده سازی بهینه ای برای درج در یک هرم بیشینه با فرمت d-ary ارائه دهید و پیچیدگی زمانی آن را بر اساس d و n تحلیل کنید.

حل. عملیات درج به کار رفته در هرم عادی در این هرم همچنان کارایی دارد. الگوریتم worst-case مشابه هرم عادی $O(h)$ می‌باشد. با این تفاوت که در این روش ارتفاع درخت $O(\log n / \log d)$ است. \triangleright

(موفق باشید :)