به نام خدا

ساختمان داده‌ها و الگوریتم‌ها (۴۰۲۵۴)

دانشگاه صنعتی شریف — تمرین سوم

مدرس: مهدی صفرنژاد — انتشار: ۳ اردیبهشت ۱۴۰۱

درخت

سؤالات را با دقت بخوانید و روی همه آن‌ها وقت بگذارید. تمرین‌های تئوری تحویل گرفته نمی‌شوند اما از آن‌ها سؤالات کوییز مشخص می‌شود. بنابراین روی سؤالات به خوبی فکر کنید و در کلاس‌های حل تمرین مربوطه شرکت کنید.

سؤال ۱. به سوالات زیر در رابطه با درخت Red-black پاسخ دهید:

آ. در چه شرایطی بهتر است از درخت Red-black به جای AVL استفاده کنیم؟

ب. ارتفاع سیاه (black height) را برای هر گره درونی در این نوع درخت تعریف کنید. نشان دهید در هر درخت Red-black با ریشه x حداقل $n = 2^{bh(x)} - 1$ گره داخلی وجود دارد ($bh(x)$ ارتفاع سیاه گره x است.).

سؤال ۲. درخت دودویی جست‌وجو را طوری تغییر دهید تا بتوان kامین عدد را در $O(log(n))$ بدست آورد. این تغییر بر روی کدام بخش از درخت (حافظه، زمان ساخت، ...) اعمال شده است؟ مرتبه تغییر را مشخص کنید.

سؤال ۳. رابطه‌ای بازگشتی برای تعداد د.د.ج های مختلف، که میتوان با اعداد $a_1 < a_2 < a_3 < ... < a_n$ ساخت را بیابید.

حال فرض کنید که یک د.د.ج ثابت داریم. رابطه‌ای بازگشتی برای تعداد دنباله‌های متفاوت از اعضای این درخت بیابید که در صورت درج آن‌ها به ترتیب، می‌توان این د.د.ج را به‌دست آورد.

سؤال ۴. یک د.د.ج با ارتفاع h در نظر بگیرید. نشان دهید با شروع از هر راس میتوان در $O(h+k)$ ، k عنصر بعدی آن را یافت.

سؤال ۵. آرایه‌ای شامل n عدد متمایز داریم. همچنین عدد k کوچک‌تر از n داده شده است. عددی را خوب می نامیم اگر از همه ی اعداد سمت چپ خود، و از حداقل k عدد سمت راستش بزرگتر باشد. الگوریتمی از $O(nlog(n))$ ارائه دهید که تعداد اعداد خوب در این آرایه را بیابد.

سؤال ۶. می‌خواهیم عمل $Tree-Enumerate(x, a, b)$ را بر روی زیردرخت دودویی جست‌وجو به ریشه‌ی x بنویسیم به طوری که تمام کلیدهایی را پیدا کند که مقدار آن‌ها بین a و b است. یک الگوریتم کارا از $\mathcal{O}(h+m)$ برای این کار ارائه دهید (h ارتفاع درخت و m تعداد جواب است).

سؤال ۷. داده‌ساختار «صف اولویت میانه» یا « $MeanPriorityQueue$ » شامل n عنصر مجزاست و اعمال زیر، روی این داده‌ساختار قابل اجرا می‌باشند:

- درج یک عنصر، در بدترین حالت در $\mathcal{O}(lg\ n)$
- حذف / دریافت عنصر میانه، در بدترین حالت در $\mathcal{O}(lg\ n)$

با استفاده از هرم، این داده‌ساختار را طراحی کنید و نحوه‌ی انجام اعمال فوق را دقیقاً توضیح دهید و تحلیل نمایید.

سؤال ۸. فرض کنید H_1 و H_2 دو هرم بیشینه هستند که به صورت درختی (و نه با آرایه) پیاده‌سازی شده‌اند؛ بنابراین شما به ریشه‌ی هر هرم و به دو فرزند و پدر هر عنصر دست‌رسی دارید. الگوریتم $Merge-Heap(H_1, H_2)$ را به‌طور کامل بنویسید تا در زمان $\mathcal{O}(lg\ n)$ این دو هرم را در هم ادغام کنید و آن‌ها را به یک هرم جدید تبدیل نمایید. در صورت نیاز، در الگوریتم خود می‌توانید از اعمال تعریف شده بر روی هرم‌ها استفاده کنید. (توجه داشته باشید که ارتفاع درخت‌های H_1 و H_2 نیز از $\mathcal{O}(lg\ n)$ می‌باشد.)

سؤال ۹. فرض کنید که علاوه بر نشانگرهای فرزند، می‌خواهیم تعداد کل گره‌های زیردرخت یک ریشه خاص را با عنوان کلید نگهداری کنیم.

آ. تابع BSTInsert را برای نگهداری صحیح کلیدها تغییر دهید. آیا زمان اجرا تغییر می‌کند؟

ب. با داشتن مقادیر کلید، شبه‌کدی برای تابع BSTKeyLessThan(T،k) بنویسید که یک درخت T و یک عدد k را می‌گیرد و تعداد کلیدهای T را که کمتر از k هستند برمی‌گرداند. بدترین زمان اجرای این تابع چیست؟

سؤال ۱۰. به سوالات زیر درباره درخت اِی‌وی‌اِل و جستجوی دودویی پاسخ دهید.

آ. فرض کنید ۷ عنصر را در یک BST درج می‌کنید. ارتفاع‌های ممکن درخت پس از درج چیست؟ (در مورد ترتیب‌های مختلف درج عناصر فکر کنید).

ب. اگر ۷ عنصر را در درخت AVL قرار دهید، ارتفاع درخت چقدر خواهد بود؟

پ. با داشتن یک درخت جستجوی دودویی، توضیح دهید که چگونه می‌توانید آن را به یک درخت AVL با حداکثر زمان $O(n\log(n))$ تبدیل کنید. بهترین زمان اجرای الگوریتم چه خواهد بود؟

ت. آیا بین ارتفاع درخت AVL و حداقل یا حداکثر تعداد گره‌های آن رابطه وجود دارد؟

سؤال ۱۱. در این سوال، در مورد درخت جستجوی سه‌گانه (TST) صحبت خواهیم کرد. درخت‌های جستجوی سه‌گانه شبیه درخت‌های جستجوی دودویی هستند، اما به جای داشتن فقط ۲ نشانگر (چپ و راست)، ۳ اشاره‌گر (چپ، وسط و راست) دارند. درخت T یک درخت سه‌گانه است که در آن هر گره حاوی یک نقطه به شکل (x, y) برای $x, y \in Z$ است. علاوه بر این، هیچ دو نقطه‌ای در T نمی‌تواند یک مقدار x یا یک مقدار y داشته باشند. ویژگی‌های زیر برای هر گره u با کلید (x, y) در هر TST برقرار است:

- هر نقطه (x_L, y_L) در زیردرخت سمت چپ u دارای $x_L < x$ و $y_L < y$ است.
- هر نقطه (x_M, y_M) در زیردرخت وسط u دارای $x_M > x$ و $y_M < y$ است.
- هر نقطه (x_R, y_R) در زیردرخت سمت راست u دارای $x_R > x$ و $y_R > y$ است.

می‌توانید فرض کنید که برای هر گره v در درخت T، تعداد گره‌هایی را که به زیردرخت v تعلق دارند، در زمان $O(1)$ محاسبه کنید.

آ. اثبات یا رد: برای هر مجموعه‌ای از n نقطه متمایز (که در آن هیچ دو نقطه با مختصات x یا مختصات y مشترک نیستند)، یک درخت جستجوی سه‌گانه T در این n نقطه با $h(T) = O(f(n))$ وجود دارد:
- وقتی $f(n) = n$،
- وقتی $f(n) = \log n$.

ب. فرض کنید به شما یک نقطه (x', y') و یک درخت جستجوی سه‌گانه T داده شده است که شامل n نقطه و ارتفاع h است. شما می خواهید تعیین کنید که آیا (x', y') در T موجود است یا خیر.
یک الگوریتم کارآمد برای حل این سوال طراحی کنید، درستی الگوریتم را ثابت کنید و زمان اجرای آن را بر حسب n و همچنین بر حسب h تحلیل کنید.

پ. حال، فرض کنید با یک درخت جستجوی سه‌گانه T با n گره و ارتفاع h و یک نقطه (x', y')، می‌خواهید تعداد نقاط (x, y) را در درخت تعیین کنید به طوری که $x' \leq x$ و $y \leq y'$.
یک الگوریتم بازگشتی ارائه دهید که با شروع از ریشه درخت، این نقاط را جستجو می‌کند. در هر مرحله بازگشتی در یک گره u از T، الگوریتم شما در بدترین حالت به چند فرزند از u نیاز دارد؟
بدترین حالت زمان اجرا $U(h)$ الگوریتم خود را بر حسب h تحلیل کنید. بدترین حالت زمان اجرا بر حسب n چیست (زمانی که h بتواند دلخواه باشد)؟ اگر درخت کاملاً متوازن باشد و $h = \log_3 n$، زمان اجرا بر حسب n چقدر خواهد بود؟

سؤال ۱۲. در یک درخت دودویی جستجوی متوازن با n عنصر، بدترین پیچیدگی زمانی برای گزارش تمام عناصر در بازه $[a, b]$ چقدر است؟ با توجه به این که تعداد عناصر گزارش شده، k تا است.

سؤال ۱۳. فرض کنید دو عنصر a و b از یک درخت دودویی جستجو داده شده است. الگوریتمی پیشنهاد دهید که بزرگترین عنصر در مسیر دو عنصر داده شده را بیابد. توجه داشته باشید که مسیر بین دو عدد همواره خود اعداد را هم شامل می‌شود.
(پیچیدگی زمانی باید O(n) باشد که n ارتفاع درخت است.)

سؤال ۱۴. فرض کنید آرایه arr[] را در اختیار دارید که شامل اعداد صحیح غیرتکراری است. آرایه temp[] را گونه‌ای بسازید که temp[i] برابر تعداد عناصر سمت راست عنصر arr[i] باشد به گونه‌ای که از این عدد کوچیک‌تر باشند. توضیح دهید ساخت این آرایه با درخت AVL به چه صورت است و مرتبه‌ی زمانی آن را حساب کنید.

سؤال ۱۵. برای عدد طبیعی $k \geq 1$، روش مرتب سازی سریع رندوم ترکیبی، الگوریتمی است که برای $n > k$ از مرتب سازی سریع رندوم و برای $n \leq k$ از مرتب سازی درجی استفاده می کند. به ازای چه مقادیری از k این الگوریتم در $O(nlogn)$ کار می کند؟

موفق باشید