به نام خدا

دانشگاه تهران، دانشکده مهندسی برق و کامپیوتر
تحلیل و طراحی الگوریتم‌ها

تمرین کتبی اول

موعد تحویل: دوشنبه ۹ اسفند ۱۴۰۰، ساعت ۲۳:۵۹

طراح: حسام اسداله‌زاده، asadzadeh.hesam@ut.ac.ir

---

۱. (۱۴ نمره) برای مسائل زیر، با استفاده از روش تقسیم و حل، الگوریتم‌هایی با زمان اجرای $O(\log n)$ ارائه دهید.

(آ) آرایه‌ی مرتب‌شدهٔ A (به صورت صعودی و به طول n) از اعداد صحیح متمایز را در نظر بگیرید. الگوریتمی بنویسید که اندیس i را بیابد به طوری که $A[i] = i$ باشد.

(ب) فرض کنید A یک آرایه‌ی یک‌بعدی با اندازه‌ی n از اعداد طبیعی متمایز باشد. $A[i]$ را ماکسیمال گوییم اگر از هر دو خانهٔ کناری‌اش (در صورت وجود) کوچک‌تر نباشد. الگوریتمی بنویسید که یک عنصر ماکسیمال را برگرداند. (ممکن است بیش از یک عنصر ماکسیمال وجود داشته باشد که ارائه فقط یکی از آن‌ها به عنوان جواب کافی خواهد بود).

۲. (۱۶ نمره) آرایه‌ی مرتب‌شده A (به صورت صعودی و به طول n) را در نظر بگیرید. به زیر دنبالهٔ متوالی $\{A[i], A[i+1], ..., A[i+j]\}$ یک دنباله با سطح جدایی k می‌نامیم اگر و فقط اگر اختلاف هر دو عضو متوالی این دنباله حداکثر برابر k باشد.

(آ) با استفاده از روش تقسیم و حل، الگوریتمی با زمان اجرای $O(n \log n)$ ارائه دهید که بزرگ‌ترین زیردنباله با سطح جدایی k را در آرایه بیابد.

(ب) سعی کنید الگوریتمی با زمان اجرای $O(n)$ برای حل مسئله ارائه دهید.

۳. (۱۵ نمره) بُرنا زمان تولد و مرگ تعداد زیادی از مشاهیر تاریخ از قرون وسطی تا پایان جنگ جهانی دوم را در دفتر خود نوشته. او می‌خواهد بداند کدام دو نفر در لیست او بیشترین هم‌پوشانی را در طول عمر خود داشته‌اند. برای این کار او آرایه‌ای از سه‌تایی‌های مرتب تشکیل داده که هر عضو آن به شکل مقابل است: $(birth, death, name)$. الگوریتمی با زمان اجرای $O(n \log n)$ ارائه دهید که این دو نفر را برای برنا بیابد. (در واقع باید بیشینه $min\{A[i].death, A[j].death\} - max\{A[i].birth, A[j].birth\}$ را به ازای $i \neq j$ بیابید). مثال:

$$n = 4, A = [(1889, 1945, AdolfHitler), (1451, 1506, ChristopherColumbus), (1878, 1953, JosephStalin),$$
$$(1483, 1546, MartinLuther)] \rightarrow answer = (AdolfHitler, JosephStalin)$$

۴. (۱۵ نمره) روی میزی n جعبه وجود دارد که داخل هر یک از آنها مقداری پول موجود است. جعبه‌ها از 0 تا $n-1$ شماره‌گذاری شده‌اند و می‌دانیم که داخل جعبهٔ iاُم، $M[i]$ دلار قرار داده شده است. جعبه‌ها بین m نفر تقسیم خواهند شد. هرکدام از این m نفر تعدادی از جعبه‌ها را به صورت متوالی دریافت می‌کنند (به طور مثال جعبه‌های ۴ و ۵ و ۶ را می‌توان به یک نفر اختصاص داد ولی حق دادن جعبه‌های ۴ و ۶ به یک نفر وجود ندارد). هر طوری که جعبه‌ها را بین افراد تقسیم کنیم، یک نفر هست که مجموع پولی که دارد از بقیه بیشتر خواهد بود. این مقدار را بین همهٔ افرازهای ممکن، $MaxMoney$ می‌نامیم. می‌خواهیم کمترین $MaxMoney$ را به ازای تمام روش‌هایی که می‌توان جعبه‌ها را بین افراد تقسیم کرد به دست آوریم. فرض کنید مقدار کل پول موجود حداکثر برابر k دلار است. از آنجایی که تعداد جعبه‌ها بسیار زیاد است، الگوریتمی ارائه دهید که مسئلهٔ بالا را در $O(n \log k)$ حل کند.

۵. (۲۰ نمره) درخت دودویی، درختی‌ست که هریک از رئوس آن دقیقا **دو** یا **صفر** فرزند دارند. درخت دودویی کامل با ارتفاع k، درخت دودویی‌ای است که تا ارتفاع $k-1$، همه رئوس آن دقیقا دو فرزند دارند و راس‌های ارتفاع kاُم هیچ فرزندی ندارند. توجه کنید که k بزرگ‌ترین عدد ممکنی‌ست که $2^k - 1$ کوچک‌تر از n باشد ($n \geq 2^k - 1 \geq 1$). حال رئوس درخت دودویی را از 1 تا $2^k - 1$ شماره‌گذاری می‌کنیم طوری که فرزندان راس iاُم دو راس $2i$ و $2i-1$ باشند.

حال n نقطه روی صفحه داریم به طوری که هیچ ۳ نقطه‌ای روی یک خط نمی‌باشند. الگوریتمی با زمان اجرای $O(n \log^2 n)$ ارائه دهید که تعدادی از این نقاط را طوری به هم وصل کند که یک درخت دودویی کامل با ارتفاع k تشکیل شود. (یال‌های درخت نباید یکدیگر را قطع کنند)

۶. (۲۰ نمره) ضرب ماتریس‌ها و بردارها یکی از عملیات مهم و رایج در الگوریتم‌های هوش مصنوعی و یادگیری عمیق به شمار می‌روند و انجام این محاسبات در زمان بهینه، بسیار حائز اهمیت می‌باشند. ماتریس‌های $M_0, M_1, M_2, \ldots$ به صورت زیر تعریف می‌شوند:

- M_0 ماتریس $[1]$ است.
- برای هر $k > 0$، ماتریس M_k یک ماتریس $2^k \times 2^k$ است که به صورت زیر تعریف می‌شود:

$$M_k = \begin{bmatrix} M_{k-1} & M_{k-1} \\ M_{k-1} & -M_{k-1} \end{bmatrix}$$

بردار ستونی، ماتریسی با ابعاد $n \times 1$ است و شامل n درایه در ستونی واحد است. ضرب ماتریس در بردار ستونی به شکل زیر تعریف می‌شود:

$$\mathbf{x} = \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} \rightarrow \mathbf{Ax} = \begin{bmatrix} a_{11} & a_{12} & \dots & a_{1n} \\ a_{21} & a_{22} & \dots & a_{2n} \\ \vdots & \vdots & \ddots & \vdots \\ a_{m1} & a_{m2} & \dots & a_{mn} \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ \vdots \\ x_n \end{bmatrix} = \begin{bmatrix} a_{11}x_1 + a_{12}x_2 + \cdots + a_{1n}x_n \\ a_{21}x_1 + a_{22}x_2 + \cdots + a_{2n}x_n \\ \vdots \\ a_{m1}x_1 + a_{m2}x_2 + \cdots + a_{mn}x_n \end{bmatrix}.$$

بردار ستونی v با تعداد اعضای $n = 2^k$ را در نظر بگیرید. ثابت کنید که حاصل ضرب $M_k v$ را می‌توان در زمان $O(n \log n)$ محاسبه کرد. فرض کنید محاسبات جمع و ضرب در زمان $O(1)$ انجام می‌شوند. (پاسخ نهایی یک بردار ستونی با اندازه $2^k \times 1$ خواهد بود)