



مسئله‌ی ۱. شمارنده [۱۲ نمره]

مسئله شمارنده دودویی را با این تفاوت در نظر بگیرید که هزینه تغییر i امین بیت به جای ۱ برابر i باشد. ثابت کنید هزینه سرشکن عمل افزایش همچنان $O(1)$ است.

مسئله‌ی ۲. وارون [۱۳ نمره]

آرایه‌ی A از n عدد دل‌خواه داده شده است. فرض کنید عملیات $reverse(i, j)$ ($1 \leq i < j \leq n$)، زیرآرایه‌ی $A[i..j]$ را معکوس می‌کند. با چه مرتبه زمانی‌ای استفاده از این عملیات می‌توان آرایه‌ی A را مرتب کرد؟ دلیل خود را ذکر کنید.

مسئله‌ی ۳. تصادفی [۲۰ نمره]

در الگوریتم مرتب‌سازی سریع تصادفی، هربار یک جدا کننده به صورت تصادفی یکنواخت انتخاب می‌شود. درخت این الگوریتم را تصور کنید که در آن در هر گره یک جدا کننده ذخیره می‌شود. یک گره را خوب می‌نامیم اگر جدا کننده آن باعث شود نسبت دو درخت چپ و راست حداقل $\frac{1}{4}$ و حداکثر $\frac{3}{4}$ باشد. در غیر این صورت آن را گره بد می‌نامیم.

(الف) نشان دهید در یک مسیر از ریشه به برگ تعداد گره‌های خوب از مرتبه $O(\log n)$ است.

(ب) به ازای دو گره خوب متوالی در یک مسیر از ریشه به برگ نشان دهید متوسط تعداد گره‌های بد $O(1)$ است.

(ج) نشان دهید متوسط طول یک مسیر دلخواه از ریشه به برگ از مرتبه $O(\log n)$ است.

(د) با توجه به قسمت ج بدون استدلال اضافی آیا می‌توان گفت ارتفاع درخت از مرتبه $O(\log n)$ است؟ توضیح دهید.

مسئله‌ی ۴. رابطه بازگشتی [۲۰ نمره]

روابط بازگشتی را حل کنید.

$$T(n) = 2T\left(\frac{n}{4}\right) + n \log n \quad (\text{الف})$$

$$T(n) = 25T\left(\frac{n}{5}\right) + n^2 \quad (\text{ب})$$

$$T(n) = \sqrt{n}T(\sqrt{n}) + 100n \quad (\text{ج})$$

مسئله‌ی ۵. مرتب‌سازی [۱۵ نمره]

یک الگوریتم با مرتبه زمانی $O(n)$ برای مرتب‌سازی تعدادی عدد طبیعی که در مجموع n رقم دارند ارائه دهید.

مسئله‌ی ۶. آرایه [۲۰ نمره]

آرایه $A[1 \dots n]$ از اعداد طبیعی متمایز داده شده است. هدف ما پیدا کردن بزرگترین k است به طوری که $1 \leq i \leq n - k$ وجود داشته باشد که $A[i] < A[i + k]$. برای مثال آرایه $A[1 \dots 7]$ که به این شکل تعریف شده است را در نظر بگیرید:

$$A[1] = 14 \quad A[2] = 6 \quad A[3] = 8 \quad A[4] = 1 \quad A[5] = 12 \quad A[6] = 7 \quad A[7] = 5$$

در این آرایه جواب ۴ است زیرا $A[2] = 6$ و $A[2 + 4] = 7$.

الف) الگوریتمی از مرتبه زمانی $O(n)$ ارائه کنید که برای هر t کمینه اعداد در بین $A[1 \dots t]$ را محاسبه کند و آن را در $MA[t]$ ذخیره کند. (در حقیقت $MA[t] = \min_{i=1}^t A[i]$)

ب) الگوریتمی از مرتبه زمانی $O(n \log n)$ ارائه دهید تا ماکزیمم k خواسته شده را پیدا کند.