



مسئله‌ی ۱. میانگان [۱۵ نمره]

آرایه A شامل n عدد مختلف است. حال می‌خواهیم آرایه B را به این صورت پر کنیم که به ازای هر i ، $B[i]$ برابر با میانه‌ی اعداد $A[1]$ تا $A[i]$ باشد. الگوریتمی از مرتبه‌ی $O(n \log n)$ برای این مسئله ارائه دهید.

مسئله‌ی ۲. قرمز-سیاه و چرخش [۲۵ نمره]

- (آ) آیا می‌توان گره‌های هر درخت دودویی جست‌وجویی که ارتفاع آن حداکثر $2 \log n$ است را با رنگ‌های قرمز و سیاه و طبق قواعد درخت قرمز و سیاه رنگ‌آمیزی کرد؟ (۶ نمره)
- (ب) نشان دهید با عمل‌های چرخش به راست و چرخش به چپ می‌توان هر درخت دودویی جست‌وجویی را به هر درخت دودویی جست‌وجوی دیگری تبدیل کرد. (۶ نمره)
- (ج) نشان دهید هر د.د.ج را با $O(n \log n)$ چرخش می‌توان به هر د.د.ج دیگر تبدیل کرد. (۱۳ نمره)

مسئله‌ی ۳. درخت دودویی جستجو [۲۰ نمره]

اعداد صحیح x_1, \dots, x_n را در یک درخت دودویی جستجو با ارتفاع h ذخیره کرده‌ایم. فرض کنید هزینه جستجوی x_i (تعداد مقایسه‌های لازم در درخت برای پیدا کردن x_i) برابر c_i باشد. می‌دانیم $\sum_{i=1}^n c_i = O(n \log n)$ است. درستی یا نادرستی گزاره‌های زیر با ذکر دلیل (مثال نقض یا اثبات) مشخص کنید.

$$h = O(\log n) \quad (\text{آ})$$

$$h = O(\sqrt{n \log n}) \quad (\text{ب})$$

$$\text{(ج) می‌توان مثالی زد که } h = \Omega(n) \text{ باشد.}$$

$$h = \Omega(\sqrt{n}) \quad (\text{د})$$

مسئله‌ی ۴. جانمایی [۲۰ نمره]

فرض کنید آرایه A شامل یک جایگشت اعداد $1, \dots, n$ باشد. مشخص کنید در کد زیر اگر به جای X دستورهای `if` و `while` بگذاریم آیا عناصر حتماً مرتب می‌شوند؟ دلیل خود را برای هر دو حالت بیان کنید.

```
sort(A) {
    for i = 1 to n
        X (A[i] < i)
        swap(A[i], A[A[i]])
}
```

مسئله ۵. پیش زوجیت پویا [۲۰ نمره]

در این مسئله قصد داریم داده ساختاری برای بررسی زوجیت یک بازه از عناصر در یک آرایه‌ی بیتی طراحی کنیم. این داده ساختار باید از اعمال زیر پشتیبانی کند:

- $\text{initialize}(n)$: ساخت آرایه‌ای به اندازه‌ی n و مقداردهی عناصر آن با صفر.

- $\text{flip}(i)$: وارون کردن مقدار بیت i ام.

- $\text{parity}(i)$: بازگرداندن زوجیت عناصر تا عنصر i ام.

فرض می‌کنیم که عمل initialize در زمان $\mathcal{O}(n)$ انجام می‌شود.

(آ) داده ساختاری طراحی کنید که اعمال $\text{flip}(i)$ و $\text{parity}(i)$ در زمان $\mathcal{O}(\log n)$ انجام شود. (۱۲ نمره)

(ب) داده ساختاری طراحی کنید که اعمال $\text{flip}(i)$ و $\text{parity}(i)$ را در زمان $\mathcal{O}(\log n / \log \log n)$ دهد. همچنین می‌توانید پیش‌پردازی به روال initialize اضافه کنید به طوری که مرتبه‌ی زمانی آن را تغییر ندهد (راهنمایی: می‌توانید از ایده‌ی B-Tree استفاده کنید). (۸ نمره)