

به نام خدا

امتحان اول درس طراحی و تحلیل الگوریتم‌ها (زمستان ۹۹)

مدت امتحان: ۹۰ دقیقه

امتحان شامل سه بخش است و از هر بخش شما باید فقط و فقط یک سوال را حل کنید. برای انتخاب سوال مربوط به خود به توضیحات بالای هر بخش توجه کنید. برای انتخاب سوال خود فرض کنید شماره دانشجویی شما $ABCDEFGHI$ باشد و مقادیر A تا I را با توجه به شماره دانشجویی خود در هر بخش جایگزین کنید. برای مثال اگر شماره دانشجویی شما 894456123 است مقدار $A = 8$ و مقدار $H = 2$ خواهد بود. لطفا در بالای برگه جواب حتما علاوه بر نام و نام خانوادگی شماره دانشجویی و شماره سوال مربوط به خود را بنویسید

بخش اول (۲۰ نمره): با توجه به شماره دانشجویی خود سوال زیر را حل کنید

۱. در این سوال هدف اجرای الگوریتم quick sort است. فرض کنید آرایه A به طول 9 شامل شماره دانشجویی شما به عنوان ورودی داده شده است. به بیانی دقیق‌تر $A[i]$ برابر i -امین رقم شماره دانشجویی شما (از چپ به راست) است. الگوریتم quick sort را به منظور مرتب کردن آرایه A اجرا کنید. به طور دقیق تمام زیرمسئله‌ها را مشخص کنید و جواب آنها را مشخص کنید. فرض کنید از کد زیر برای تقسیم آرایه استفاده می‌کنیم:

Algorithm 1 partition($A, first, last$)

```
pivot = A[last]
i = first
for j = first to last - 1 do
    if A[j] < pivot then
        swap(A[i], A[j])
        i = i + 1
    end if
end for
swap(A[i], A[last])
return i
```

بخش دوم (۴۰ نمره): با توجه به شماره دانشجویی خود مقدار H و I را محاسبه کنید. اگر باقیمانده $I + H$ بر 2 به ترتیب 0، 1 بود به ترتیب سوال 2 یا 3 را حل کنید.

۲. ماتریس جادویی، ماتریسی است که در هر خانه آن عدد 0 یا 1 نوشته شده است و هم‌چنین اعداد هر سطر غیر نزولی هستند. یک ماتریس جادویی 1000×1000 داریم و تنها راه دسترسی به این ماتریس استفاده از تابع $getValue(i, j)$ است که مقدار خانه‌ی (i, j) را برمی‌گرداند.

الگوریتمی طراحی کنید که با حداکثر 15000 بار صدا زدن تابع $getValue(i, j)$ سمت چپ‌ترین ستونی را پیدا کند که در آن حداقل یک عدد 1 ظاهر شده است. نشان دهید که الگوریتم شما در هیچ حالتی بیش از 15000 بار تابع مورد نظر را صدا نمی‌زند.

۳. درخت جستجوی دودویی، درختی است که دارای خواص زیر است:

- به هر راس یک عدد نسبت داده شده است.
- به غیر از ریشه، هر راس یک پدر دارد. همچنین هر راس حداکثر دو فرزند دارد که آنها را فرزند سمت چپ و راست می‌نامیم.
- به ازای هر راس، عدد نسبت داده شده به آن از تمام اعداد زیردرخت سمت چپ آن بزرگتر است و از تمام اعداد زیر درخت سمت راست آن بزرگتر نیست.

یک درخت جستجوی دودویی با 1001 راس، ارتفاع 20 و ریشه r شده است. تنها راه دسترسی به این درخت استفاده از تابع `getNodeInfo(i)` است که مقادیر زیر را به ما برمی‌گرداند:

value → the value stored at node i .
size → the number of nodes in the subtree rooted at node i .
left → the left child of node i . it might be *null*.
right → the right child of node i . It might be *null*.

الگوریتمی طراحی کنید که با حداکثر 40 بار صدا زدن تابع `getNodeInfo(i)` میانه اعداد درخت ورودی را پیدا کند. نشان دهید که الگوریتم شما در هیچ حالتی بیش از 40 بار تابع مورد نظر را صدا نمی‌زند.

بخش سوم (۴۰ نمره): با توجه به شماره دانشجویی خود مقدار I و G را محاسبه کنید. اگر باقیمانده $I + G$ بر 2 به ترتیب 0، 1 بود به ترتیب سوال 4 یا 5 را حل کنید.

۴. آرایه A به طول n از اعداد صحیح داده شده است. دقت کنید که اعداد این آرایه می‌توانند مثبت یا منفی باشند. همچنین بازه‌ی $[l, r]$ نیز به عنوان ورودی داده شده است. الگوریتمی با زمان اجرای $O(n \times \log(n))$ طراحی کنید که تعداد زوج‌های (i, j) را پیدا کند که $i < j$ و $A[j] - A[i] \in [l, r]$. شبه کد مربوط به الگوریتم خود را بنویسید.

۵. آرایه A به طول n از اعداد صحیح متمایز داده شده است. الگوریتمی با زمان اجرای $O(n \times \log(n))$ طراحی کنید که برای هر خانه آرایه تعداد خانه‌های سمت راست آن که از آن کوچکتر هستند را حساب کند. شبه کد مربوط به الگوریتم خود را بنویسید.