ساختمان داده - دکتر آبین

امیرحسین منصوری - ۹۹۲۴۳۰۶۹ - تمرین سری ۱

سوال ۱ - (a

نادرست - به ازای $f(n) = 2n$ و $g(n) = n$ ، داریم $f(n) \in O(g(n))$. زیرا به ازای $c = 3$ و 0 و $n_0 = 0$ و $n \ge n_0$:

$$f(n) \le cg(n) \Rightarrow 2n \le 3n$$

اما برای $2^{f(n)}$ و $2^{g(n)}$ ، داریم:

$$\lim_{n\to\infty} \frac{2^{f(n)}}{2^{g(n)}} = \lim_{n\to\infty} \frac{2^{2n}}{2^{n}} = \infty$$

که با خواص Big-O تناقض دارد. بنابراین عبارت داده شده نادرست است.

سوال ۱ - (b

تابعی که از $o(f(n))$ است را $g(n)$ در نظر می‌گیریم. می‌دانیم به ازای یک c و n_0 و به ازای همه $n > n_0$:

$$0 \le g(n) < cf(n) \quad (n > n_0)$$

بنابراین داریم:

$$f(n) \le f(n) + g(n) \le (c+1)f(n) \quad (n > n_0)$$

و بنابراین به ازای همه $n > n_0$ ، $f(n) + g(n)$ از $\Theta(f(n))$ است.

سوال ۱ - (c

درست - با توجه به فرض سوال می‌دانیم c, n_0 وجود دارد که

$$f(n) \le cg(n) \quad (n \ge n_0)$$
$$\Rightarrow log(f(n)) \le log(cg(n)) \Rightarrow log(f(n)) \le log(c) + log(g(n))$$

می‌دانیم چون $log(g(n)) \ge 1$ ، پس $g(n)$ نمی‌تواند نزولی باشد. پس به ازای n_0' ای:

$$log(c) \le log(g(n)) \quad (n \ge n_0')$$
$$\Rightarrow log(c) + log(g(n)) \le 2log(g(n)) \quad (n \ge max(n_0,\ n_0'))$$
$$\Rightarrow log(f(n)) \le 2log(g(n)) \quad (n \ge max(n_0,\ n_0'))$$

بنابراین طبق تعریف، $log(f(n))$ از $O(log(g(n)))$ است.

سوال ۲

حلقه while خط ۹، به اندازه $[log_5 i]$ اجرا می‌شود. همچنین حلقه for خط ۶، i بار اجرا می‌شود. بنابراین، به ازای هر بار اجرای حلقه for خط ۴، خط ۱۰ به اندازه $i \times [log_5 i]$ اجرا می‌شود. همچنین حلقه for خط ۴، که در آن مقدار i تغییر می‌کند، $[log_5 n]$ بار اجرا می‌شود، و در آن i مقدار های زیر را می‌گیرد:

$$n,\ \left[\frac{n}{5}\right], \left[\frac{n}{5^2}\right], \dots, \left[\frac{n}{5^{[log_5 n]-1}}\right]$$

بنابراین تعداد اجراهای خط ۱۰ را می‌توان به صورت زیر نوشت:

$$[log_5 n] \times n + ([log_5 n] - 1) \times \left[\frac{n}{5}\right] + ([log_5 n] - 2) \times \left[\frac{n}{5^2}\right] + \dots$$
$$+ ([log_5 n] - [log_5 n] + 1) \times \left[\frac{n}{5^{[log_5 n]-1}}\right]$$
$$= \sum_{i=0}^{[log_5 n]-1} ([log_5 n] - i) \times \left[\frac{n}{5^i}\right]$$

برای به دست آوردن پیچیدگی زمانی تابع، می‌دانیم:

$$\sum_{i=0}^{[log_5 n]-1} ([log_5 n] - i) \times \left[\frac{n}{5^i} \right] \leq [log_5 n] \times [log_5 n] \times n$$

پس می‌توان گفت تابع از $O(nlog^2 n)$ است.

سوال ۳ - a)

هر بار اجرای کامل حلقه for خط ۴، شامل n+1 بار چک کردن شرط حلقه، n بار عمل ++j، و n بار عمل ++x است. بنابراین این حلقه در اجرای کامل خودش 3n+1 حرکت انجام می‌دهد. همچنین هر بار اجرای حلقه for خط ۳، شامل n+1 بار چک کردن شرط حلقه، n بار عمل ++i، و $n + n(3n+1) = 3n^2$ حرکت برای اجرای دستورات داخل حلقه است. بنابراین در اجرای کامل حلقه خط ۳، $3n^2 + 3n + 1$ حرکت انجام می‌شود. همچنین برای اجرای خط ۱ و ۲، هر کدام ۱ حرکت انجام می‌شود. درنتیجه این الگوریتم $3n^2 + 3n + 3$ حرکت انجام می‌دهد. بنابراین مرتبه زمانی آن، $O(n^2)$ است.

سوال ۳ - b)

هر بار اجرای کامل حلقه خط ۴، شامل $j + 1$ بار چک کردن شرط حلقه، j بار عمل ++x، و j بار عمل ++k است. بنابراین در اجرای کامل این حلقه، $3j + 1$ حرکت انجام می‌شود.

همچنین هر بار اجرای کامل حلقه خط ۳، شامل $[log_2 n] + 2$ بار چک‌کردن شرط حلقه و $[log_2 n] + 1$ بار عمل j*=2 است. همچنین بدنه این حلقه به ازای مقادیر مخلتف j، به اندازه زیر اجرا می‌شود:

$$(3n + 1) + (3\left[\frac{n}{2^1}\right] + 1) + (3\left[\frac{n}{2^2}\right] + 1) + \ldots + (3\left[\frac{n}{2^{[log_2 n]+1}}\right] + 1)$$

$$= \sum_{j=0}^{[log_2 n]+1} (3\left[\frac{n}{2^j}\right] + 1)$$

بنابراین، در اجرای کامل حلقه خط ۳، به تعداد

$$\sum_{j=0}^{[log_2 n]+1} (3\left[\frac{n}{2^j}\right] + 1) + 2[log_2 n] + 3$$

حرکت انجام می‌شود.

همچنین هر بار اجرای کامل حلقه خط ۲، شامل $[log_2 n] + 2$ بار چک‌کردن شرط حلقه و $[log_2 n] + 1$ بار عمل i*=2، و $[log_2 n] + 1$ بار اجرای بدنه حلقه است. بنابراین اجرای کامل حلقه خط ۲، به تعداد

$$([log_2 n] + 1)(\sum_{j=0}^{[log_2 n]+1} (3\left[\frac{n}{2^j}\right] + 1) + 2[log_2 n] + 3) + 2[log_2 n] + 3$$

عمل انجام می‌شود. با احتساب خط ۱، در کل الگوریتم به تعداد

$$T(n) = ([log_2 n] + 1)(\sum_{j=0}^{[log_2 n]+1} (3\left[\frac{n}{2^j}\right] + 1) + 2[log_2 n] + 3) + 2[log_2 n] + 6$$

عمل انجام می‌دهد.

می‌دانیم:

$$\sum_{j=0}^{[log_2 n]+1} (3\left[\frac{n}{2^j}\right] + 1) \leq ([log_2 n] + 2)(3n + 1) = 3n[log_2 n] + [log_2 n] + 6n + 1$$

$$\Rightarrow T(n) \leq ([log_2 n] + 1)(3n[log_2 n] + 3[log_2 n] + 6n + 4) + 2[log_2 n] + 6$$

$$\Rightarrow T(n) \leq (3n[log_2 n]^2 + 3[log_2 n]^2 + 9n[log_2 n] + 9[log_2 n] + 6n + 10$$

در نتیجه الگوریتم از مرتبه $O(nlog^2 n)$ است.

سوال ۴

تعداد همه زوج‌مرتب‌های (i,j) که در آن i و j متمایز باشند برابر $C(n,2) = \frac{n(n-1)}{2}$ است. با توجه به این‌که احتمال وارونگی بودن یک زوج مرتب ۰.۵ است، می‌توان فرض کرد که نصف این زوج‌مرتب‌ها وارونگی هستند. بنابراین به طور میانگین، $\frac{n^2-n}{4} = \frac{n(n-1)}{4}$ وارونگی در آرایه وجود دارد.

سوال ۵

سوال نادرست است - الگوریتم داده شده مقدار min و max را درست محاسبه نمی‌کند.