

تمرین شماره 1

موعد تحویل 29 فروردین ساعت 12 ظهر

مشورت با دیگران مجاز نیست

پاسخ همه سوالات تشریحی باشد

:1

برای هریک از توابع زیر، رشد مجانبی تابع را با نماد θ مشخص کنید:

$$\begin{array}{ll}
 n^2 \log n - n\sqrt{n} + 10 \log_n^{10} & \theta(n^2 \log n) \\
 \log^2 n^5 + 10 \log n^{10} & \theta(\log^2 n) \\
 \sqrt{n} \log^3 n + \log^4 n & \theta(\sqrt{n} \log^3 n) \\
 n \times \sum_{i=0}^n 5^{-i} & \theta(n)
 \end{array}$$

$\theta(1)$

2: درستی یا نادرستی روابط زیر را نشان دهید:

$$(a) \quad n^{2/3} \in o(n^2) \quad \text{درست}$$

$$(b) \quad 10^{1000}n \in O(n \log n) \quad \text{درست}$$

$$(c) \quad 5000n \in \omega(n) \quad \text{نادرست}$$

$$\lim_{n \rightarrow \infty} \frac{n^{2/3}}{n^2} = 0$$

$$10^{1000}n \leq n \log n, \quad n \geq 2^{1000}$$

$$\lim_{n \rightarrow \infty} \frac{5000n}{n} = 5000$$

:3

رابطه بازگشتی زیر را حل کنید.

$$\begin{cases} T(1) = 0 \\ T(n) = 4T(n/2) + n^2 \log n \end{cases} \quad n > 1$$

$$T(n) \leq 4T\left(\frac{n}{2}\right) + \theta\left(n^{\log_2 4} \times (\log n)^k\right)$$

$$\Rightarrow T(n) \leq \theta\left(n^{\log_2 4} (\log n)^{k+1}\right)$$

نتیجه:

:4

رشد مجانبی توابع زیر را محاسبه نمایید.

$$T(n) = 2T(n/8) + \sqrt[3]{n}$$

$$a=2, b=8, n^{\log_8 2} = n^{\frac{1}{3}} = O(\sqrt[3]{n})$$

$$T(n) = O(n^{\frac{1}{3}} \log n) \quad \text{طبق حالت دوم معینه اصل}$$

:5

۱. با استفاده از قضیه اصلی روابط بازگشتی زیر را حل کنید:

$$a=4, b=2, n^{\log_2 4} = n^2 \Rightarrow n = O(n^{2-1}) \quad T(n) = 4T\left(\frac{n}{2}\right) + n$$

$$T(n) = O(n^2) \quad \text{طبق حالت اول معینه اصل}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n^2$$

$$a=4, b=2, n^{\log_2 4} = n^2 = O(n^2)$$

$$T(n) = n^2 \log n \quad \text{طبق حالت دوم معینه اصل}$$

:6

قطعه برنامه‌های زیر دارای چه زمان اجرایی (big-o) می‌باشند.

(a) `sum=0;`
`for (i=0; i<n; i++)`
`sum ++;`
`for (j=0; j<n; j++)`
`sum ++;`

(b) `sum=0;`
`for (i=0; i<n/2; i++)`
`for (j=0; j<n/4; j++)`
`sum ++;`

(c) `sum=0;`
`for (i=1; i<n; i++)`
`for (j=1; j<i*i; j++)`
`for (k=1; j<k; k++)`
`if (j%i==0)`
`sum ++;`

Handwritten notes for (a): $\sum_{i=0}^n 1 + \sum_{j=0}^n 1 = O(n)$

Handwritten notes for (b): $\sum_{i=0}^n \sum_{j=0}^{n/4} 1 = O(n^2)$

Handwritten notes for (c): $\sum_{i=1}^n \sum_{j=1}^{i^2} \sum_{k=1}^j 1 = \sum_{i=1}^n \sum_{j=1}^{i^2} j = O(n^5)$

Handwritten notes for (c): $t = \log n$

:7

پیچیدگی زمانی قطعه کد زیر چیست؟

`for (i=1; i<n; i++)`
`for (j=1; j<n; j=2*j)`
`for (k=0; k<j; k++)`
`x=x+1;`

Handwritten notes: $\sum_{i=1}^{n-1} \sum_{t=1}^{\log n - 1} \sum_{k=0}^{2^t - 1} 1 =$

Handwritten notes: $\sum_{i=1}^{n-1} \sum_{t=1}^{\log n - 1} 2^t = \sum_{i=1}^{n-1} \frac{2^{\log n - 1} - 1}{2 - 1} = \sum_{i=1}^{n-1} (n-2) = (n-1)(n-2)$

:8

رابطه بازگشتی زیر را حل کنید:

$$T(n) = 3T\left(\frac{n}{3} + 5\right) + \frac{n}{2}$$

نکته: عدد 5 تأثیری در مرتبه ندارد

$$T(n) = 3T\left(\frac{n}{3} + \frac{n}{3}\right)$$

$$T(n) = O(n \log n)$$

از قضیه اصل قوت دوم

$$S(n) = T(n + \alpha) = 3T\left(\frac{n + \alpha}{3} + \delta\right) + \frac{n + \alpha}{2} = 3T\left(\frac{n}{3} + \frac{\alpha}{3} + \delta\right) + \frac{n + \alpha}{2}$$

$$= 3S\left(\frac{n}{3} + \frac{\alpha}{3} + \delta - \frac{\alpha}{3}\right) + \frac{n + \alpha}{2} = 3S\left(\frac{n}{3} + \delta - \frac{\alpha}{3}\right) + \frac{n + \alpha}{2}$$

$$S(n) = 3S\left(\frac{n}{3}\right) + \frac{n + \alpha}{2} = 3S\left(\frac{n}{3}\right) + \frac{n}{2} + \frac{\alpha}{2}$$

حال نه مقینه اصل ریاضی را در نظر

9: اگر در الگوریتم ادغامی به جای نصف کردن آرایه آن را به سه قسمت مساوی تقسیم کنیم پیچیدگی آن را بدست آورید.

$$T(n) = 3T(\frac{n}{3}) + n$$

$$T(n) = O(n \log n)$$

10: آرایه صعودی S را با عناصر متمایز داریم. یک الگوریتم $O(\log n)$ طراحی کنید که آیا k وجود دارد که $S(k) = k$ ؟ *Binary search*

$$m = \frac{1+n}{2} \quad \text{if } A(m) = m \quad \checkmark$$

if $A(m) < m \Rightarrow$ search in $[m+1, n]$ *جواب آرایه صعودی است*
if $A(m) > m \Rightarrow$ search in $[1, m-1]$

11: یک ماتریس در نظر بگیرید که سطرهایش از چپ به راست صعودی و ستونهایش هم از بالا به پایین صعودی هستند. برای پیدا کردن یک کلید داده شده در آن بهترین الگوریتم را آرایه دهید.

کامپیوت از گدنه مت جیب دین شری کنیم نه حد اکثر ۱-۲ مقایسه لازم دارد.

۹۵	۷۵	۸۵	۹۷
۴۸	۷۸	۸۸	۹۸
۸۹	۹۱	۱۰۰	۱۰۲
۹۵	۹۹	۱۰۹	۱۱۹

هدف مقایسه ۹۳

12:

پس از اجرای قطعه کد زیر، k از چه مرتبه ای است؟

k = 0

for(i = 0; i < n; i = i + 1)

for(j = 1; j < n; j = j + i)

k = k + 1

$n^2 \lg n$ (۴)

n^2 (۳)

$n \lg n$ (۲)

n (۱)

?

:13

 $k = 0$ $\text{for}(i = 1; i < n; i++)$ $\text{for}(j = 1; j < i; j = j++)$ $k = k + i + j$ n^4 (4)پس از اجرای قطعه که زیر، k از چه مرتبه‌ای است؟

بزرگم n اعداد $\Rightarrow \sum_{i=1}^{n-1} (i(i-1) + i(i-1)) = \frac{2}{3} \sum_{i=1}^{n-1} i(i-1) = \frac{2}{3} \sum_{i=1}^{n-1} (i^2 - i) = \frac{2}{3} \left(\sum_{i=1}^{n-1} i^2 - \sum_{i=1}^{n-1} i \right) = \frac{2}{3} \left(\frac{(n-1)n(2n-1)}{6} - \frac{(n-1)n}{2} \right) = \frac{2}{3} \left(\frac{(n-1)n(2n-1-3)}{6} \right) = \frac{2}{3} \left(\frac{(n-1)n(n-2)}{6} \right) = \frac{(n-1)n(n-2)}{9}$

n^3 (3) n^2 (2) n (1)

:14

با توجه به رابطه بازگشتی زیر، $T(n)$ از چه مرتبه‌ای است؟

$$\begin{cases} T(2^n) = T(2^{n-1}) + \theta(n^2) \\ T(1) = \theta(1) \end{cases}$$

 n^2 (2)
 $(\lg n)^2$ (4) $\lg n$ (1)
 $n^{\lg 2}$ (3)

$$\begin{aligned} T(2^n) &= T(2^{n-1}) + \theta(n^2) + \theta(n^2) + \dots \\ &= T(1) + \theta(1^2) + \dots + \theta(n^2) \\ &= \theta(n^3) \Rightarrow T(n) = \theta((\log_2 n)^3) \end{aligned}$$

:15

فرض کنید می‌خواهیم یک آرایه ۱۰ عنصری را با استفاده از الگوریتم مرتب‌سازی ادغامی، مرتب کنیم. چند مورد از موارد زیر صحیح است؟

(I) تعداد فراخوانی تابع merge در این الگوریتم ۱۹ است.

(II) تعداد فراخوانی‌های بازگشتی برابر با ۹ است.

(III) حداکثر تعداد مقایسه‌ها بین عناصر آرایه ۲۵ تا است.

(4) هیچ کدام

(3) ۳ مورد

(2) ۲ مورد

(1) ۱ مورد ✓

$$T(n) = n \log n - (n-1)$$

$$n=1 \quad T(n) \leq 78$$

$$\sum_{i=1}^n \frac{1}{i^2} = 1 + \frac{1}{4} + \frac{1}{9} + \dots$$

$\theta(1) + \theta(1) + \dots = \theta(n)$

:16

$$\sum_{i=1}^n \frac{1}{i^2}$$

مرتبه زمانی محاسبه رابطه روبه رو کدام است؟

$\theta(n)$ (۲)
 $\theta(n^2)$ (۴)

$\theta(1)$ (۱)

$\theta(n^3)$ (۳)

البته در انتخابه لنکوز بررسی (۱) درست است ؟؟؟

:17

آرایه ای به طول n داریم که $n - k$ عنصر آن مرتب و k عنصر آن نامرتب هستند. اگر k خیلی کوچک تر از n باشد آنگاه کدام الگوریتم مرتب سازی دارای بهترین کارایی برای مرتب کردن آن است؟

Selection (۴)

Insertion (۳)

Merge (۲)

Heap (۱)