دانشکده علوم ریاضی و آمار

مدرس: دکتر مجتبی رفیعی نیمسال اول ۱۴۰۰-۱۴۰۱

ساختمان داده‌ها و الگوریتم‌ها - طرح سوال جلسات ۷ تا ۲۰

مهلت تحول: *۱ آذر ۱۴۰۰* ***زمان اشتراک گذاری:*** *۱۵ آبان ۱۴۰۰*

- پاسخ‌ها باید در قالب یک سند PDF و با نام شماره دانشجویی (StudentNumber.pdf) در سامانه LMS بارگذاری شود. هر گونه فایل در قالب تصویر یا زیپ نادیده گرفته خواهد و هیچ نمره‌ای به آن تخصیص داده نخواهد شد.
- به پاسخ‌های مشابه نمره‌ای داده نمی‌شود. لذا بعد از همفکری با دوستان خود، لطفا با جملات خودتان اقدام به نگارش تکلیف نمایید.
- تمرین‌هایی که به رایانامه درس ارسال می‌شوند مورد بررسی قرار نخواهد گرفت و در نتیجه نمره‌ای هم برای ان لحاظ نمی‌شود.
- حداکثر اندازه مجاز برای فایل ارسالی 3 MB می‌باشد.
- مهلت زمانی ارسال پاسخ‌نامه ساعت ۱۱:۵۵ روز مشخص شده در مستند تمرین است و این زمان قابل تمدید نخواهد بود.
- پاسخ هر سوال می‌بایست دقیق و متناسب با سوال باشد. لذا از ذکر مطالب مبهم، نامرتبط و زاید خودداری کنید.
- حداکثر تعداد صفحات پاسخ می‌بایست ۱۰ صفحه باشد.
- در صورت استفاده از منابع خاصی برای پاسخ به سوال، نام منابع را ذکر کنید.
- پاسخ‌ها می‌توانند به طور کامل به زبان فارسی یا به طور کامل به زبان انگلیسی نوشته شوند، و لذا ترکیبی از هر دو مجاز نیست.
- در صورت نقض هر یک از موارد ذکر شده، نمره کسر خواهد شد.

$3 \rightarrow \Theta(n^2 \log n) \leftarrow$ حالت ۲

$4 \rightarrow \Theta(n^2) \leftarrow$ حالت ۳

$5 \rightarrow \Theta(n^{\log_2(7)}) \leftarrow$ حالت ۱

$6 \rightarrow \Theta(n^{\frac{1}{2}} \log n) \leftarrow$ حالت ۲

مشت ۳ تا ۶ با بررسی قضیه اصلی قابل محاسبه است:

سوال ۱

(۵ نمره) حل هر یک از رابطه‌های بازگشتی زیر را در قالب نمادهای مجانبی $\mathcal{O}$ و Ω تعیین کنید.

◁ لازم به ذکر است که هدف پیدا کردن پایین‌ترین مرز برای نماد $\mathcal{O}$ و بالاترین مرز برای نماد Ω می‌باشد.

◁ روش‌های مجاز: قضیه اصلی، حدس و استقرا، درخت بازگشت و جایگذاری.

- ۱ $T(n) = 2T(\frac{n}{2}) + n^4,$
- ۲ $T(n) = T(\frac{7n}{10}) + n,$
- ۳ $T(n) = 16T(\frac{n}{4}) + n^2,$
- ۴ $T(n) = 7T(\frac{n}{3}) + n^2,$
- ۵ $T(n) = 7T(\frac{n}{2}) + n^2,$
- ۶ $T(n) = 2T(\frac{n}{4}) + \sqrt{n},$
- ۷ $T(n) = T(n-2) + n^2,$

یادآوری قضیه اصلی: برای $a \geq 1$, $b > 1$ و $c = \log_b a$:

حالت ۱: اگر $f(n) = O(n^{c-\varepsilon})$ آنگاه $T(n) = \Theta(n^c)$،

حالت ۲: اگر $f(n) = \Theta(n^c \log^k n)$ آنگاه $T(n) = \Theta(n^c \log^{k+1} n)$،

حالت ۳: اگر $f(n) = \Omega(n^{c+\varepsilon})$ and $af(n/b) \leq h f(n)$, where $0 < h < 1$ آنگاه $T(n) = \Theta(f(n))$.

① $a=2, b=2, c=1, f(n)=n^4 \longrightarrow$ حالت ۳: $n^4 = \Omega(n^{1+\varepsilon})$ and

$2 f(\frac{n}{2}) \leq h f(n)$

$2 \frac{n^4}{16} \leq h n^4$

$\frac{1}{8} \leq h$

پس داریم $T(n) = \Theta(n^4)$.

---

② $a=1, b=\frac{10}{7}, c=0, f(n)=n \longrightarrow$ حالت ۳: $n = \Omega(n^{\varepsilon})$ and

$f(\frac{7n}{10}) \leq h f(n)$

$\frac{7}{10} n \leq h n$

$\frac{7}{10} \leq h$ ✓

پس داریم $T(n) = \Theta(n)$.

---

⑦ از طریق جایگذاری

$T(n) = T(n-2) + n^2$

$\hookrightarrow T(n-4) + (n-2)^2$

$\hookrightarrow T(n-6) + (n-4)^2$

$\vdots$

$T(n) = \sum_{i=0}^{n/2} (n-2i)^2 = \Theta(n^3)$

سوال ۲

(۵ نمره) شبه کد مربوط به نسخه بازگشتی و نسخه تکراری الگوریتم جستجوی دودویی را نوشته و سپس پیچیدگی زمانی آن را تحلیل و درستی آن را اثبات کنید.

◁ برای تحلیل پیچیدگی نسخه بازگشتی از فرم کلی پیچیدگی زمانی مربوط به رویکرد تقسیم و غلبه استفاده کرده و بخش‌های آن را به طور دقیق مشخص کرده و سپس اقدام به حل آن نمایید.

(مهم) * فرض می‌کنیم آرایه $A[1..n]$ مرتب است.

◁ برای اثبات درستی، به طور دقیق گام‌های ناوردایی حلقه (در صورت نیاز) و گام‌های استقرا را تعیین کنید.

الگوریتم بازگشتی

```
RBinarySearch (A[1..n], a, b, v)
1. if (a > b) then
2.     return Null
3. m = ⌊(a+b)/2⌋
4. if (A[m] == v) then
5.     return m
6. if (A[m] < v) then
7.     return RBinarySearch(A[1..n], m+1, b, v)
8. else
9.     return RBinarySearch(A[1..n], a, m-1, v)
```

الگوریتم تکراری

```
IBinarySearch (A[1..n], v)
1. a = 1, b = n
2. while (a ≤ b) do
3.     m = ⌊(a+b)/2⌋
4.     if (A[m] == v) then
5.         return m
6.     if (A[m] < v) then
7.         a = m + 1
8.     if (A[m] > v) then
9.         b = m - 1
```

پیچیدگی:

$$T(n) = \underset{1}{a}\,T(n/\underset{2}{b}) + \underbrace{D(n)}_{O(1)} + \underset{O(1)}{C(n)}$$

$$T(n) = T(n/2) + O(1)$$

حل با قضیه اصلی، $a=1$, $b=2$, $c=\log_2 1=0$

حالت 1 ← $\times$ $f(n)=O(n^{c-\varepsilon})$

حالت 2 ← $\checkmark$ $f(n)=\Theta(n^0 \log^{k=0} n)$ ← $T(n)=\Theta(n^0\log n) = \Theta(\log n)$

حالت 3 ← $\times$ $f(n)=\Omega(n^{c+\varepsilon})$

پس پیچیدگی آن $O(\log n)$ است.

ایده اثبات درستی ←
برای الگوریتم بازگشتی: اثبات استقرایی،
برای الگوریتم تکراری: نا وردایی حلقه روی خطوط ۲ تا ۹

سوال ۳

(۵ نمره) رابطه بازگشتی $T(n) = 4T(\frac{n}{2}) + n^2 \log n$ را در نظر گرفته و به سوالات زیر پاسخ دهید.

- آیا قضیه اصلی قابل اعمال روی رابطه بازگشتی فوق می‌باشد؟ حالت‌های مختلف را بررسی و هر یک از آنها را رد یا تایید نمایید.
- برای رابطه بازگشتی فوق، یک حد مجانبی بالا پیدا کنید.

◁ در قسمت اول، ذکر جواب بله/خیر کفایت نمی‌کند و می‌بایست حالت‌های مختلف را تحلیل کنید.
◁ در قسمت دوم، پایین‌ترین مرز مجانبی بالا مد نظر است.

حالت‌های قضیه اصلی در پاسخ مربوط به سؤال ۱ را جهت یادآوری مشاهده کنید.

$a=4,\ b=2,\ c=\log_2 4=2,\ f(n)=n^2\log n\ ,\ \underline{a\geq 1,\ b>1}$

حالت ۱: $n^2\log n = O(n^{2-\varepsilon})$ ×

حالت ۲: $n^2\log n = \Theta(n^2\log^k n)$, where $k=1$, ✓ s.t. $T(n)=\Theta(n^2\log^2 n)$

حالت ۳: $n^2\log n = \Omega(n^{2+\varepsilon})$ ×

در نتیجه، قضیه اصلی قابل اعمال بوده و شرایط حالت ۲ را ارضا می‌کند. بنابراین پیچیدگی مجانبی آن $\Theta(n^2\log^2 n)$ است و با توجه به تعریف نماد Θ، حد بالای مجانبی $O(n^2\log^2 n)$ را داریم.

یادآوری:

$$f(n)=\Theta(g(n)) \iff f(n)=O(g(n)) \text{ and } f(n)=\Omega(g(n))$$

نکته: حالت دوم که در کلاس تشریح شد با حالت دوم کتاب کمی متفاوت است و امکان بررسی کلاس بزرگتری از روابط بازگشتی را می‌دهد. این تمرین نکته ذکر شده را به خوبی نشان می‌دهد.

حالت ۲ کتاب: $f(n)=\Theta(n^c) \longrightarrow T(n)=\Theta(n^c\log n)$

حالت ۲ تشریح شده در کلاس: $f(n)=\Theta(n^c\log^k n) \longrightarrow T(n)=\Theta(n^c\log^{k+1} n)$ برای $k\geq 0$

سوال ۴

(۵ نمره) شرط زیر را که برای حالت سوم قضیه اصلی بیان گردید، در نظر بگیرید.

Regularity condition

$$af(\frac{n}{b}) \leq hf(n)), \quad \text{for some constant, } h < ۱ \qquad (*)$$

← به این شرط اطلاق می‌شود.

تابع $f(n)$ای را پیشنهاد دهید که شرایط زیر برای آن برقرار باشد:

- شرط فوق (*) برای آن برقرار نباشد،
- رابطه $f(n) = \Omega(n^{c+\epsilon})$ برای آن برقرار باشد.

◁ علیرغم فرضیاتی که به طور معمول برای تابع $f(n)$، مبنی بر صعودی بودن آن لحاظ شد، تابع پیشنهادی شما می‌تواند صعودی نباشد (با اینحال شرط متناوب نبودن تابع همواره می‌بایست برقرار باشد، چرا که در غیر اینصورت مفاهیم مجانبی برای آنها قابل تعریف نیست).

$$f(n) = \begin{cases} 3n + 2^{3n} & \exists i \in N \text{ s.t } 2^i = n \\ 3n & \text{o.w.} \end{cases}$$

برای تابع بالا می‌دانیم هر عددی که ۳ برابر توان ۲ یک عدد باشد:

$$f(n) = 3n < 2^n + n = f(n/3) \qquad (**)$$

حال رابطه (*) را برای f پیشنهادی بررسی می‌کنیم، فرض کنید $a = 1$ و $b = 3$ باشد به عبارتی رابطه بازگشتی ما

$$T(n) = T(n/3) + f(n)$$

باشد. طبق رابطه (*) :

$$af(n/b) \leq h f(n)$$

$$f(n/3) \leq h f(n)$$

برای هر n به اندازه کافی بزرگ، اما برای nهای قالب معرفی شده در بالا (**) عکس این رابطه برقرار است.

با اینحال به وضوح رابطه $f(n) = \Omega(n^{c+\epsilon})$ برقرار است، جاییکه $c = \log_3 1 = 0$ ، $\epsilon = 1$.

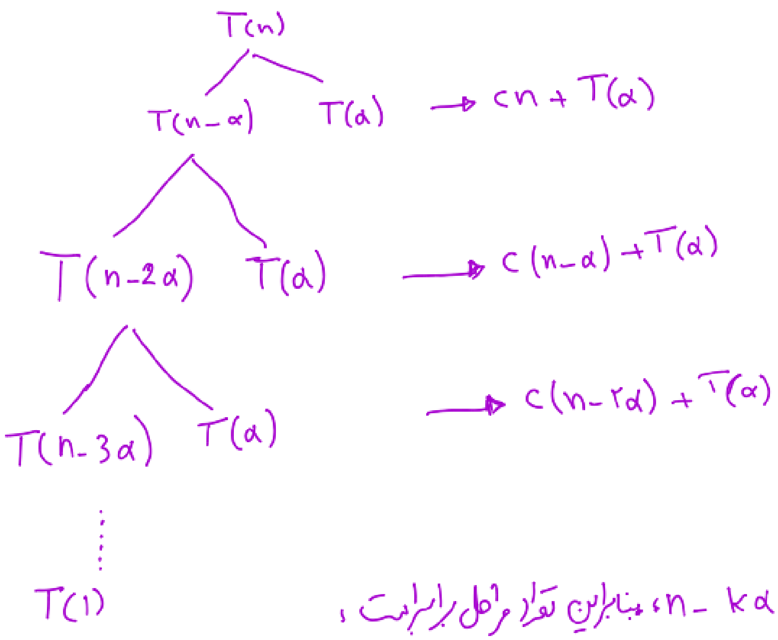
سوال ۵

(۵ نمره) با استفاده از درخت بازگشت، توابع بازگشتی زیر را حل کنید.

◁ بیان پایین‌ترین کران بالا (یا بالاترین کران پایین) کفایت می‌کند و نیاز به ارایه حل دقیق نیست.

1. $T(n) = T(\alpha n) + T\big((1-\alpha)n\big) + cn$, where $0 < \alpha < \frac{1}{2}$ and $c > 0$,
2. $T(n) = T(n-\alpha) + T(\alpha) + cn$, where $1 \le \alpha$ and $c > 0$.

جواب ۲ :



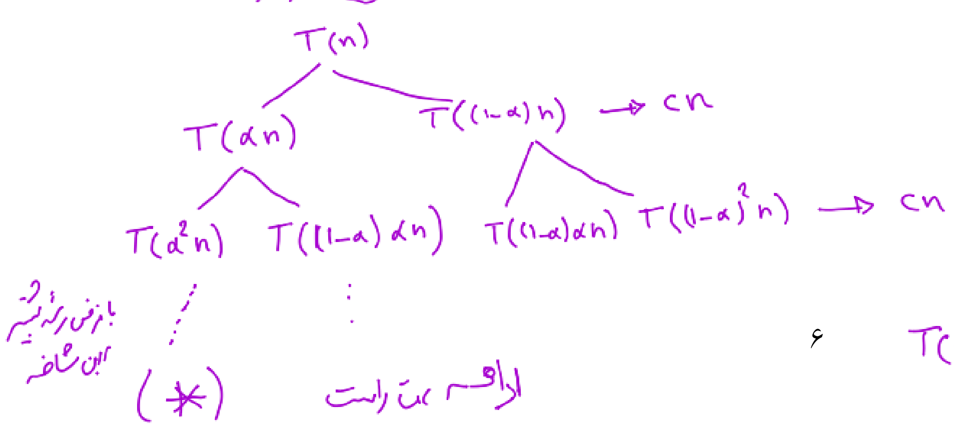
تا جایی ادامه می‌دهیم که $n - k\alpha = 1$، بنابراین تعداد مراحل برابر است،

$$n-1 = k\alpha \Rightarrow k = \frac{n-1}{\alpha}$$

$$\underbrace{\sum_{k=0}^{\frac{n-1}{\alpha}} T(\alpha)}_{O(1)} + \sum_{k=0}^{\frac{n-1}{\alpha}} c(n-k\alpha) = O(1) + \sum_{k=0}^{\frac{n-1}{\alpha}} cn - \alpha \sum_{k=0}^{\frac{n-1}{\alpha}} k = cn\left(\frac{n-1}{\alpha}+1\right) - \left(\frac{\frac{n-1}{\alpha}\cdot\frac{n-1}{\alpha}+1}{2}\right)\alpha$$

$$= O(n^2)$$

جواب ۱ : یکی از شاخه‌ها متناسب با مقدار α رشد بیشتری در مقایسه با شاخه دیگر دارد (آن شاخه‌ای که α بزرگتری دارد).



فرض کنید $\alpha = \frac{1}{b}$ باشد، داریم: (*)

$$\left(\frac{1}{b}\right)^k n = 1 \Rightarrow \frac{n}{b^k} = 1$$

$$\Rightarrow n = b^k \Rightarrow k = \log_b n$$

بنابراین :

$$T(n) = \underbrace{2^{\log_b n} T(1)} + cn(\log_b n) = \Theta(n \log n)$$

1 $0 \le \underbrace{c_1(f(n)+g(n)) \le \max(f(n),g(n))}_{(*)} \le c_2(f(n)+g(n))$, $(**)$: $\max(f(n),g(n)) \le c_2(f(n)+g(n))$

$(*) \rightarrow c_1 = 1/2$

$(**) \rightarrow c_2 = 1$

سوال ۶

(۵ نمره) فرض کنید $f(n)$ و $g(n)$ دو تابع مثبت باشند، بررسی کنید که آیا روابط زیر برقرار است یا خیر؟

- ۱ $\max(f(n), g(n)) = \theta((f(n) + g(n)))$, ✓ $\longrightarrow$ به حل بالای صفحه رجوع کنید
- ۲ $f(n) + g(n) = \theta\Big(\min(f(n), g(n))\Big)$ ✗ $\longrightarrow$ $n + n^2 \neq \Theta(n)$
- ۳ $f(n) = \mathcal{O}(g(n)) \;\; \rightarrow \;\; g(n) = \mathcal{O}(f(n))$ ✗ $\longrightarrow$ $n = O(n^2)$ but $n^2 \neq O(n)$
- ۴ $o(g(n)) \cap \omega(g(n)) = \emptyset$,
- ۵ $f(n) = \theta(f(\frac{n}{2}))$ ✗ $\longrightarrow$ $f(n) = 2^{2n}$ then $2^{2n} \neq O(2^n)$
- ۶ $f(n) + o(f(n)) = \theta(f(n))$ ✓ $\longrightarrow$ از طریق تعریف θ و o واضح است
- ۷ $2^{n+1} = \mathcal{O}(2^n)$, ✓ $\longrightarrow$ $2 \times 2^n = O(2^n)$ طبق تعریف
- ۸ $2^{2n} = \mathcal{O}(2^n)$, ✗ $\longrightarrow$ $2^n \times 2^n \neq O(2^n)$ طبق تعریف
- ۹ $(n + a)^b = \theta(n^b)$, for any real constants a and $b > 1$,

۴

فرض کنید $f(n) \in \big(o(g(n)) \cap \omega(g(n))\big)$ باشد

پس طبق تعریف o داریم:

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = 0$$

و از سوی دیگر، طبق تعریف ω داریم:

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

به تناقض می‌رسیم، پس هیچ تابع fای نمی‌تواند عضو آن باشد و این مجموعه تهی است.

۹ $(n+a)^b = \Theta(n^b) \Leftrightarrow \underbrace{(n+a)^b = O(n^b)}_{(*)}$ and

$\underbrace{(n+a)^b = \Omega(n^b)}_{(**)}$

$(*)$: $c = 2^b$, $n_0 \geqslant 2a$, $\forall n \geqslant n_0$ $(n+a)^b \leqslant (2n)^b = cn^b$

و در نتیجه $(n+a)^b = O(n^b)$

$(\overset{1}{**})$: $c = \frac{1}{2}$, $n_0 = \dfrac{-a}{1 - \frac{1}{2^{\frac{1}{b}}}}$, $\forall n \geqslant n_0 = \dfrac{-a}{1 - \frac{1}{2^{\frac{1}{b}}}}$ $\Leftrightarrow$ $n - \dfrac{n}{2^{1/b}} \geqslant -a$

$\Leftrightarrow$ $n + a \geqslant \dfrac{n}{2^{1/b}} = \left(\frac{1}{2}\right)^{\frac{1}{b}} n$

دو طرف را به توان b می‌رسانیم $\Leftrightarrow$ $(n+a)^b \geqslant \underbrace{\frac{1}{2}}_{c} n^b$

$(\overset{2}{**})$: $(x+y)^n = \binom{n}{0} x^n y^0 + \binom{n}{1} x^{n-1} y^1 + \cdots + \binom{n}{n-1} x^1 y^{n-1} + \binom{n}{n} x^0 y^n$

به وضوح قابل محاسبه است، در اینجا y ثابت a است و x هم معادل n است.

سوال ۷

(۵ نمره) روابط مجانبی زیر را بررسی و درستی/نادرستی آنها را تحلیل کنید.

- $2^{n+1} = \mathcal{O}(2^n)$,
- $2^{2n} = \mathcal{O}(2^n)$,
- $(n+a)^b = \theta(n^b)$, for any real constants a and $b > 1$.

در سوال قبل حل شد.

سه گزینه آخر سوال ۶ با سوال ۷ مشترک است که اشتباه شده است.

سوال ۸

(۵ نمره) روابط مجانبی را بررسی و سلول‌های جدول را با بله/خیر پر کنید.

$f(n)$	$g(n)$	$\mathcal{O}$	o	Ω	ω	θ
$\log^k n$	n^ϵ	✓	✓	✗	✗	✗
2^n	$2^{\frac{n}{2}}$	✗	✗	✓	✓	✗
$\sqrt{n}$	$n^{\sin n}$	–	–	–	–	–
$n^{\log c}$	$c^{\log n}$	✓	✗	✓	✗	✓

Table 1: $k \geq 1,\ \epsilon > 0,\ c > 1$