

Data Structures and algorithms

Instructor : Dr.Amin Eskandari

Head-Ta's : Hamid Namjoo manesh, Amir Hossein Hemati , Ali Mojahed
Ta's : Amir Mohammad Asadjoo , Padina razmjooi , Armin Janfaza , Alireza Heydari , Mahdi Torabi ,
Maryam Ghorbani , Fatemeh Keshavarz , MohammadReza Fasli , Abolfazl Tadrissi

Week 2.5 Answers

پاسخنامه تکالیف هفته 02 (پک تکمیلی)

یاد آوری :

توابعی به فرم $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ که a و b ثابت هستند و $b > 1, a > 1$ را می توان با استفاده از قضیه اصلی (Master theorem) حل کرد .

$$(۱) \text{ اگر } T(n) \in \theta(n^{\log_b^a}) \leftarrow n^{\log_b^a} > f(n)$$

$$(۲) \text{ اگر } T(n) \in \theta(f(n)) \leftarrow n^{\log_b^a} < f(n)$$

$$(۳) \text{ اگر } T(n) \in \theta(f(n)\log(n)) \leftarrow n^{\log_b^a} = f(n) \text{ (هم مرتبه)}$$

(۴) هیچ یک از سه حالت بالا رخ ندهد $\leftarrow$ با قضیه اصلی قابل حل نیست.

پاسخ سوال اول :

$$۱) T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$a = 2, b = 2, f(n) = n \rightarrow n^{\log_2^2} = n^1 \rightarrow f(n), n^{\log_b^a}$$

طبق حالت سوم قضیه اصلی و پس از مقایسه ، هر دو هم مرتبه هستند. پس : $T(n) \in \theta(n\log n)$

$$۲) T(n) = 4T\left(\frac{n}{2}\right) + cn$$

$$a = 4, b = 2, f(n) = cn \left(\text{درجه یک با ضریب ثابت} \right) \rightarrow n^{\log_2^4} = n^2$$

$$f(n) \rightarrow cn \text{ مرتبه } , n^{\log_b^a} \rightarrow n^2 \text{ مرتبه} \rightarrow n^2 > cn$$

طبق حالت اول قضیه اصلی و پس از مقایسه داریم : $T(n) \in \theta(n^2)$

$$۳) T(n) = 3T\left(\frac{n}{4}\right) + n$$

$$a = 3, b = 4, f(n) = n \rightarrow n^{\log_4 3} \approx n^{0.79}$$

$$f(n) \rightarrow n, n^{\log_b a} \rightarrow n^{0.79} \rightarrow n^{0.79} < n^1$$

طبق حالت دوم قضیه اصلی و پس از مقایسه داریم : $T(n) \in \theta(n)$

$$۴) T(n) = 2T\left(\frac{n}{2}\right) + O(n \log n)$$

$$a = 2, b = 2, f(n) = O(n \log n) \rightarrow n^{\log_2 2} = n^1$$

$$f(n) \rightarrow O(n \log n), n^{\log_b a} \rightarrow n$$

پس از مقایسه متوجه می شویم که با استفاده از قضیه اصلی قابل حل نیست.

برای حل این بخش نباید به سادگی فکر کنیم که چون $n \log n$ نسبت به n در یک $\log n$ ضرب شده بنابراین رشد

آن از n بیشتر است و در حالت دوم قضیه اصلی قرار می گیرد. اما چرا؟

با وجود اینکه $n^{\epsilon} \in O(n \log n)$ هست اما هیچ مقدار $\epsilon > 0$ وجود ندارد که به ازای آن (طبق تعریف)

$n^{1+\epsilon} \in O(n \log n)$ ، چون هر چقدر هم مقدار کمتری برای ϵ پیدا کنیم باز هم مقداری از n^0 وجود دارد که

برای $n > n^{\epsilon}$ داشته باشیم $n^{\epsilon} > \log n$ (با آوردن مثال یا رسم نمودار متوجه می شوید) پس با توجه به دلایل

ذکر شده نمی توان گفت هیچ یک از این دو تابع $n \log n, n$ به طور چند جمله ای بیشتر از تابع دیگر است و

همچنین هر دو روند رشد یکسانی هم ندارند. بنابر این از قضیه اصلی نمی توان استفاده کرد، اما بهترین روش

استفاده از روش درخت بازگشت می باشد.

روش درخت بازگشت به شرح زیر است :

• سطح : $n \log n$

هزینه کل $\left(\frac{n}{2}\right) \log \left(\frac{n}{2}\right) \rightarrow 2\left(\left(\frac{n}{2}\right) \log \left(\frac{n}{2}\right)\right)$: سطح 1 زیر مسئله $\frac{n}{2}$

هزینه کل $\left(\frac{n}{4}\right) \log \left(\frac{n}{4}\right) \rightarrow 4\left(\left(\frac{n}{4}\right) \log \left(\frac{n}{4}\right)\right)$: سطح 2 زیر مسئله $\frac{n}{4}$

:

$$T(n) = \theta(n \log^2 n)$$

$$\delta) T(n) = 16T\left(\frac{n}{4}\right) + n^2 \log n$$

$$a = 16, b = 4, f(n) = n^2 \log n \rightarrow n^{\log_4 16} = n^2$$

$$f(n) \rightarrow n^2 \log n, n^{\log_b a} \rightarrow n^2$$

این مورد هم همانند مورد قبل با استفاده از قضیه اصلی قابل حل نیست. برای این بخش نیز مانند بخش قبل

تحلیل کرده و سپس با استفاده از درخت بازگشت آن را حل کنید تا در نهایت به پاسخ نهایی زیر برسید :

$$T(n) = \theta(n^2 \log^2 n)$$

$$\epsilon) T(n) = T(\sqrt{n}) + \log n$$

اگر فرم این بخش را باز هم با قضیه اصلی مقایسه کنید مشخص است که با قضیه اصلی قابل حل نیست. پس

برای حل این بخش به سراغ روش تغییر متغیر می رویم:

$$n = 2^k \rightarrow k = \log n$$

$$T(2^k) = T\left(2^{\frac{k}{2}}\right) + k \rightarrow T(2^k) = S(k) \rightarrow S(k) = S\left(\frac{k}{2}\right) + k$$

$$S(k) = \theta(k), k = \log n \rightarrow T(n) = \theta(\log n)$$

$$\zeta) T(n) = 4T\left(\frac{n}{2}\right) + n^2 \log^3 n$$

$$a = 4, b = 2, f(n) = n^2 \log^3 n \rightarrow n^{\log_2^4} = n^2$$

$$f(n) \rightarrow n^2 \log^3 n, n^{\log_b^a} \rightarrow n^2$$

پس از مقایسه متوجه می شویم که این بخش نیز با استفاده قضیه اصلی قابل حل نیست. پس برای حل این

بخش نیز همانند بخش های ۴ و ۵ به سراغ درخت بازگشت می رویم تا در نهایت به پاسخ نهایی زیر برسیم :

$$T(n) = \theta(n^2 \log^4 n)$$

$$\text{۸)} \quad T(n) = 3T\left(\frac{n}{\sqrt{2}}\right) + n$$

$$a = 3, b = \sqrt{2}, f(n) = n \rightarrow n^{\log_{\sqrt{2}}^3} = \frac{\log_2^3}{\log_2^{\sqrt{2}}} = \frac{\log_2^3}{\frac{1}{2} \log_2^2} = \frac{\log_2^3}{\frac{1}{2}} = 2 \log_2^3 \approx 3.17 = n^{3.17}$$

$$f(n) \rightarrow n, n^{\log_b^a} \rightarrow n^{3.17}$$

پس طبق حالت اول قضیه اصلی و پس از مقایسه داریم : $T(n) \in \theta(n^{3.17})$

$$\text{۹)} \quad T(n) = \sqrt{2}T\left(\frac{n}{2}\right) + \log n$$

$$a = \sqrt{2}, b = 2, f(n) = \log n \rightarrow n^{\log_2^{\sqrt{2}}} = n^{\frac{1}{2} \log_2^2} = n^{\frac{1}{2}} = \sqrt{n}$$

$$f(n) \rightarrow \log n, n^{\log_b^a} \rightarrow \sqrt{n}$$

با توجه به حالت اول قضیه اصلی و اینکه $\sqrt{n} > \log n$ داریم : $T(n) = \theta(\sqrt{n})$

$$\text{۱۰)} \quad T(n) = 9T\left(\frac{n}{27}\right) + \sqrt[3]{27}$$

$$a = 9, b = 27, f(n) = \sqrt[3]{27} \rightarrow n^{\log_{27}^9} = n^{\log_{3^3}^{3^2}} = n^{\frac{2}{3} \log_3^3} = n^{\frac{2}{3}}$$

$$f(n) = 3 \rightarrow \theta(1), n^{\log_b^a} \rightarrow n^{\frac{2}{3}}$$

طبق حالت اول قضیه اصلی ، پس از مقایسه $n^{\frac{2}{3}} > 3$ پاسخ روبرو بدست می آید : $T(n) = \theta\left(n^{\frac{2}{3}}\right)$

$$\text{۱۱)} \quad T(n) = 4T\left(\frac{n}{2}\right) + n \log n$$

$$a = 4, b = 2, f(n) = n \log n \rightarrow n^{\log_2^4} = n^2$$

$$f(n) \rightarrow n \log n, n^{\log_b^a} \rightarrow n^2$$

بیایید ابتدا تفاوت این بخش را با بخش های 4 و 5 مقایسه کنیم:

طبق حالت اول قضیه اصلی، تابع $n \log n$ از n^2 به صورت چند جمله ای کوچک تر است. چرا؟

در اینجا n^2 از $n \log n$ بزرگتر است. طبق تعریف آیا $f(n) = O(n^{\log_b^a - \epsilon})$ برقرار است؟ بله، به طور مثال:

$$f(n) = n \log n = O(n^{2-1}) \leftarrow \epsilon = 1$$

چون $n \log n$ از n سریعتر رشد می کند بنابراین با حالت قضیه اصلی

مطابقت دارد. حل آن با استفاده از روش درخت بازگشت نیز برای درک بهتر در ادامه آورده می شود:

سطح 0 : $n \log n$

$$\text{سطح 1 : } 4 \left(\frac{n}{2} \right) \log \left(\frac{n}{2} \right)$$

$$\text{سطح 2 : } 16 \left(\frac{n}{4} \right) \log \left(\frac{n}{4} \right)$$

:

که ارتفاع درخت برابر است با $\log_2^n$ و در نهایت نتیجه پایانی برابر است با: $T(n) = \theta(n^2)$

$$۱۲) T(n) = 3T\left(\frac{n}{4}\right) + n^3$$

$$a = 3, b = 4, f(n) = n^3 \rightarrow n^{\log_4^3} \approx n^{0.79}$$

$$f(n) \rightarrow n^3, n^{\log_b^a} \rightarrow n^{0.79}$$

طبق حالت دوم قضیه اصلی و باتوجه به مقایسه $n^3 > n^{0.79}$ داریم: $T(n) = \theta(n^3)$

$$۱۳) T(n) = T\left(\frac{n}{3}\right) + \sqrt{n}$$

$$a = 1, b = 3, f(n) = \sqrt{n} \rightarrow n^{\log_3^1} = n^0 = 1$$

$$f(n) \rightarrow \sqrt{n}, n^{\log_b^a} \rightarrow 1$$

طبق حالت دوم قضیه اصلی و باتوجه به مقایسه $\sqrt{n} > 1$ داریم $T(n) = \theta(\sqrt{n})$

$$۱۴) T(n) = 10T\left(\frac{n}{10}\right) + 10n$$

$$a = 10, b = 10, f(n) = 10n \rightarrow n^{\log_{10}^{10}} = n^1 = n$$

$$f(n) \rightarrow 10n, n^{\log_b^a} \rightarrow n$$

طبق حالت سوم قضیه اصلی و باتوجه به هم مرتبه بودن $n = 10n$ داریم $T(n) = \theta(n)$

$$۱۵) T(n) = 64T\left(\frac{n}{8}\right) + n\sqrt{n} \log n$$

$$a = 64, b = 8, f(n) = n\sqrt{n} \log n = n^{1.5} \log n \rightarrow n^{\log_8^{64}} = n^2$$

$$f(n) \rightarrow n^{1.5} \log n, n^{\log_b^a} \rightarrow n^2$$

به این بخش هم همانند بخش ۱۱ نگاه کنید!

آیا $f(n) = O(n^{\log_b^a - \varepsilon})$ ؟ بله ، به طور مثال آیا $n^{1.5} \log n = O(n^{2-\varepsilon})$ با فرض اینکه $\varepsilon = 0.25$ یعنی

$$n^{1.5} \log n = O(n^{18.5}) \text{ برقرار است.}$$

طبق حالت اول قضیه اصلی داریم $T(n) = \theta(n^2)$

$$۱۶) T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + \log n$$

اگر فرم سوال را با قضیه اصلی مقایسه کنید ، مشخص است که با استفاده از قضیه اصلی ، این بخش قابل حل

نیست. پس به سراغ روش درخت بازگشت می رویم.

سطح 0 : $\log n$

سطح 1 : $\log\left(\frac{n}{2}\right) + \log\left(\frac{n}{4}\right) \rightarrow 2\log n$ با صرف نظر از ثابت ها

سطح 2 : $\log\left(\frac{n}{4}\right) + \log\left(\frac{n}{8}\right) + \log\left(\frac{n}{8}\right) + \log\left(\frac{n}{16}\right) \rightarrow 4\log n$

:

$$T(n) \approx n \times \log n$$

که n مجموع سری هندسی و $\log n$ عمق درخت می باشد .

در نهایت پاسخ این بخش به $T(n) = \theta(n \log n)$ ختم می شود

$$۱۷) T(n) = 5T\left(\frac{n}{5}\right) + n^2 \sqrt{\log n}$$

$$a = 5, b = 5, f(n) = n^2 \sqrt{\log n} \rightarrow n^{\log_5^5} = n$$

$$f(n) \rightarrow n^2 \sqrt{\log n}, n^{\log_b^a} \rightarrow n$$

پس از مقایسه $n^2 \sqrt{\log n} > n$ با توجه به اینکه $n^2 \sqrt{\log n}$ به طور مجانبی سریع تر از n رشد می کند، با استفاده

از حالت دوم قضیه اصلی به $T(n) = \theta(n^2 \sqrt{\log n})$ می رسمیم.

$$۱۸) T(n) = 7T\left(\frac{n}{2}\right) + \frac{n^3}{\log n}$$

$$a = 7, b = 2, f(n) = \frac{n^3}{\log n} \rightarrow n^{\log_2^7} = n^{2.81}$$

$$f(n) \rightarrow \frac{n^3}{\log n}, n^{\log_b^a} \rightarrow n^{2.81}$$

پس از مقایسه به پاسخ نهایی $T(n) = \theta\left(\frac{n^3}{\log n}\right)$ می رسمیم. اما چرا؟

وقتی $f(n)$ شامل توابع لگاریتمی باشد در مورد اعمال قضیه اصلی باید بیشتر دقت کنیم.

آیا $\Omega(n^{2.81+\varepsilon}) = \frac{n^3}{\log n}$ برقرار است؟ اگر $\varepsilon = 0.1$ باشد آنگاه n^3 از $n^{2.9}$ بزرگتر است (حتی با وجود تقسیم بر

$\log n$). برخی منابع قضیه اصلی را برای توابع $f(n)$ که شامل $\log n$ هستند، دقیق تر تعریف می کند اما در این

مثال $f(n)$ به طور قابل توجهی بزرگ تر از $n^{\log_b^a}$ است. بنابراین حالت دوم قضیه اصلی اعمال می شود.

$$۱۹) T(n) = 3T\left(\frac{n}{4}\right) + 2^n$$

$$a = 3, b = 4, f(n) = 2^n \rightarrow n^{\log_4^3} = n^{0.79}$$

$$f(n) \rightarrow 2^n, n^{\log_b^a} \rightarrow n^{0.79}$$

طبق حالت دوم قضیه اصلی و با توجه به مقایسه $2^n > n^{0.79}$ پاسخ نهایی $T(n) = \theta(2^n)$ می شود.

نکته: تابع 2^n یک تابع نمایی است که بسیار سریع تر از هر تابع توانی مانند $n^{0.79}$ رشد می کند.

$$۲۰) T(n) = 5T\left(\frac{n}{2}\right) + \theta(n^4)$$

$$a = 5, b = 2, f(n) = \theta(n^4) \rightarrow n^{\log_2^5} = n^{2.82}$$

$$f(n) \rightarrow \theta(n^4), n^{\log_b^a} \rightarrow n^{2.82}$$

پس از مقایسه ($n^4 > n^{2.82}$) به پاسخ نهایی $T(n) = \theta(n^4)$ می رسیم.

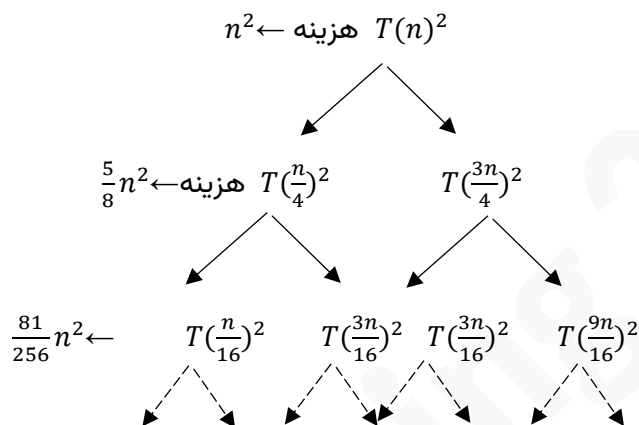
پاسخ سوال دوم:

$$۱) T(n) = T\left(\frac{n}{4}\right) + T\left(\frac{3n}{4}\right) + n^2$$

$$\text{دلیل توان } 2: \left(\frac{n}{4}\right)^2 + \left(\frac{3n}{4}\right)^2 \rightarrow n^2: \text{سطح } ۱$$

$$\text{سطح } ۲: \left(\frac{n}{16}\right)^2 + \left(\frac{3n}{16}\right)^2 + \left(\frac{3n}{16}\right)^2 + \left(\frac{9n}{16}\right)^2$$

⋮



این درخت نامتوازن است (تقسیم بندی $\frac{3n}{4}, \frac{n}{4}$) بنابراین عمق درخت توسط کوچکترین ورودی که به ۱ می رسد تعیین می شود. ارتفاع درخت $\log_{\frac{4}{3}} \frac{n}{4}$ می باشد.

$$T(n) = n^2 \sum_{i=0}^{\log_{\frac{4}{3}} \frac{n}{4}} \left(\frac{1+3^2}{4^2} \right)^i < cn^2 = \theta(n^2) \text{ (سری هندسی)}$$

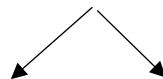
$$۲) T(n) = T\left(\frac{n}{2}\right) + T\left(\frac{n}{4}\right) + \log n$$

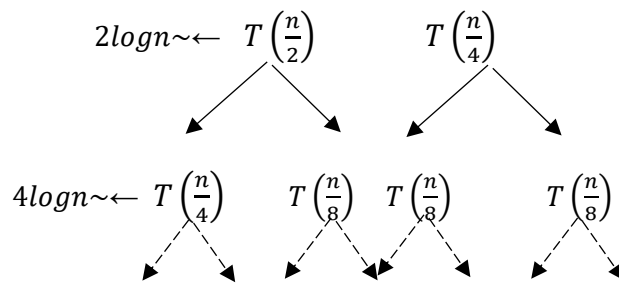
دلیل $\log : \log\left(\frac{n}{2}\right) + \log\left(\frac{n}{4}\right) \rightarrow (\log n)$ سطح ۱

سطح ۲: $\log\left(\frac{n}{4}\right) + \log\left(\frac{n}{8}\right) + \log\left(\frac{n}{8}\right) + \log\left(\frac{n}{8}\right)$

⋮

$$\log n \leftarrow \text{هزینه } T(n)$$





این درخت نیز نامتوازن است $(\frac{n}{4}, \frac{n}{2})$ بنابراین عمق درخت کوچکترین کاهش ورودی که مربوط به بزرگترین تقسیم است می باشد. ارتفاع درخت $\log_4^n$ می باشد.

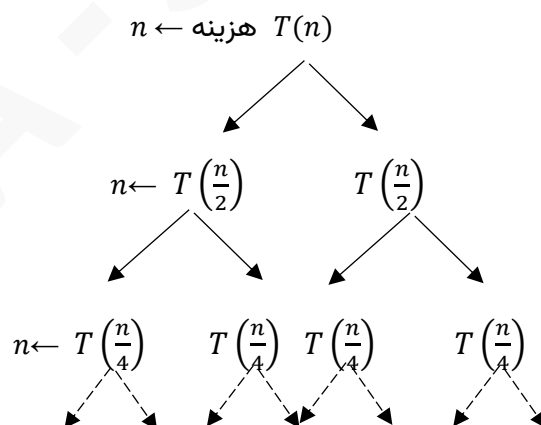
$$T(n) = \sum_{i=0}^{\log_4^n} 2^i \log n = \theta(n \log n) \text{ (سری هندسی)}$$

$$۳) T(n) = 2T\left(\frac{n}{2}\right) + n$$

$$\text{سطح ۱: } \left(\frac{n}{2}\right) + \left(\frac{n}{2}\right)$$

$$\text{سطح ۲: } \left(\frac{n}{4}\right) + \left(\frac{n}{4}\right) + \left(\frac{n}{4}\right) + \left(\frac{n}{4}\right)$$

⋮



این درخت متوازن است $(\frac{n}{2}, \frac{n}{2})$ ، بنابراین عمق آن $\log_2^n$ می باشد.

$$T(n) = \sum_{i=0}^{\log n} n = n(\log n + 1), (\text{سری هندسی}) \rightarrow \theta(n \log n)$$

$$۴) T(n) = 2T(\sqrt{n}) + O(n \log n)$$

این بخش را با استفاده از روش تغییر متغیر به صورت زیر حل می کنیم:

$$* \sqrt{n} = n^{\frac{1}{2}} \rightarrow n = 2^k \rightarrow 2^{\frac{k}{2}}$$

$$* \log n \rightarrow n = 2^k \rightarrow \log 2^k = k \log_2 2 = k$$

$$n \rightarrow \text{با روش تغییر متغیر} \quad n = 2^k$$

$$T(2^k) = 2T\left(2^{\frac{k}{2}}\right) + O(2^k \times k)$$

$$S(k) = T(2^k) \rightarrow S(k) = 2S\left(\frac{k}{2}\right) + O(2^k \times k) \leftarrow \text{فرم تابع جدید با قضیه اصلی قابل حل است.}$$

$$a = 2, b = 2, f(n) = O(2^k \times k) \rightarrow n^{\log_2 2} = n$$

$$f(n) \rightarrow k \cdot 2^k, n^{\log_b a} \rightarrow n \approx k$$

طبق حالت دوم قضیه اصلی و با توجه به مقایسه $2^k \cdot k > k$ ، $S(k) = \theta(k \cdot 2^k)$ پاسخ زیر بدست می آید :

$$n = 2^k, k = \log n \rightarrow T(n) = \theta(n \log n)$$