ساختمان داده - دکتر آبین

امیرحسین منصوری - ۹۹۲۴۳۰۶۹ - تمرین سری ۲

سوال ۱ - ii)
داریم:

$T(n) \leq 3T(\frac{n}{2}) + 1$

با استفاده از قضیه اصلی، مرتبه تابع سمت راست نامساوی را حساب می‌کنیم:

$a = 3,\ b = 2,\ f(n) = 1$

به ازای $\varepsilon = lg(3)$ ، داریم $f(n) = O(n^{lg(3)-\varepsilon}) = O(1)$. بنابراین سمت راست تابع از مرتبه $\theta(n^{lg(3)})$ است. پس $T(n) = O(n^{lg(3)})$.

سوال ۱ - iii)
با استفاده از قضیه اصلی داریم:

$a = 4,\ b = 2,\ f(n) = \frac{n}{log\ n}$

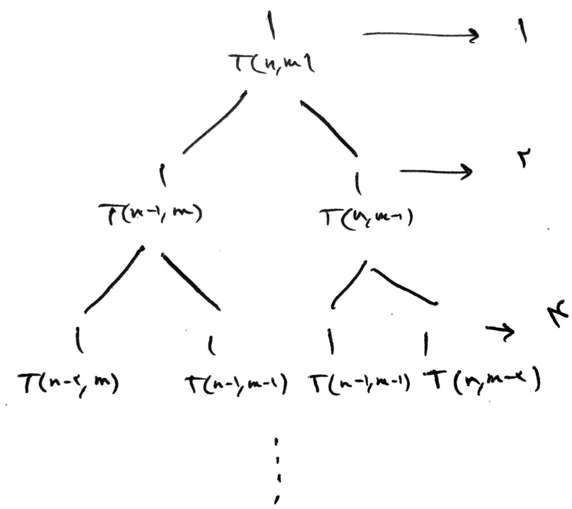
به ازای $\varepsilon = 1$ ، $f(n) = O(n^{log_2 4\ -\ 1}) = O(n)$ ، بنابراین $T(n) = O(n^{log_2 4}) = O(n^2)$

سوال ۱ - iv)
با بسط دادن تابع داریم (منظور از $C(n, r)$ ، تعداد انتخاب‌های r شی از n شی است):

$$\begin{aligned} T(n) &= T(n-1) + C(n,\ n-1) \\ &= T(n-2) + C(n,\ n-2) + C(n,\ n-1) \\ &= C(n,\ 0) + C(n,\ 1) + \dots + C(n,\ n-2) + C(n,\ n-1) \\ &= 2^n - 1 \end{aligned}$$

بنابراین $T(n) = O(2^n)$.

سوال ۱ - v)
درخت بازگشت $T(n)$ به صورت زیر است:



با توجه به درخت، تعداد سطح‌ها حداکثر برابر $max(n, m)$ است. همچنین مجموع هزینه هر سطح برابر با 2^i است که در آن i شماره سطح (با شروع از صفر) است. در نتیجه مجموع هزینه‌ها در بدترین حالت (یعنی حالتی که دقیقا $max(n, m)$ سطح داشته باشیم) برابر است با:

$$T_{worst}(n) = \sum_{i=o}^{max(n,m)} 2^i = 2^{max(n,m)+1} - 1$$

در نتیجه $T(n) = O(2^{max(n,m)})$.

سوال ۱ - vi)

طرفین را تقسیم بر n می‌کنیم:

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

تعریف می‌کنیم $S(n) = \frac{T(n)}{n}$. در نتیجه:

$$S(n) = S(\sqrt{n}) + log(n)$$

فرض می‌کنیم $n = 2^m$. بنابراین $m = log(n)$. داریم:

$$S(2^m) = S(2^{\frac{m}{2}}) + m$$

تعریف می‌کنیم $P(m) = S(2^m)$. داریم:

$$P(m) = P(\frac{m}{2}) + m$$

حال $P(m)$ را حل می‌کنیم. با استفاده از قضیه اصلی داریم:

$$a = 1,\ b = 2,\ f(m) = m$$

به ازای $\varepsilon = 1$، داریم $f(m) = \Omega(m^{log_2(1)+1}) = \Omega(m)$. بنابراین $P(m) = \theta(m)$.

بنابراین داریم:

$$P(m) = S(2^m) = \theta(m)$$
$$S(2^m) = S(n) = \theta(m) = \theta(log(n))$$
$$S(n) = \frac{T(n)}{n} \Rightarrow T(n) = nS(n)$$
$$\Rightarrow T(n) = \theta(nlog(n))$$

سوال ۱ - vii)

از جزء صحیح صرف نظر می‌کنیم. فرض می‌کنیم $n = 2^m$. بنابراین $m = log(n)$. داریم:

$$T(2^m) = 2T(2^{\frac{m}{2}}) + m$$

تعریف می‌کنیم $S(m) = T(2^m)$. بنابراین داریم:

$$S(m) = 2S(\frac{m}{2}) + m$$

با استفاده از قضیه اصلی داریم:

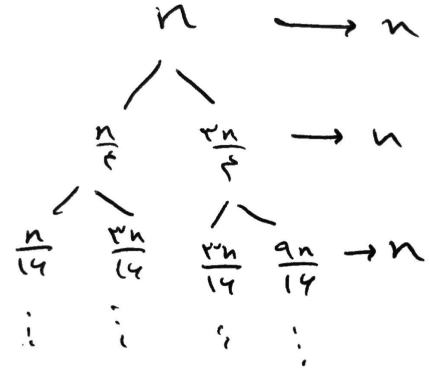
$$a = 2,\ b = 2,\ f(m) = m$$

داریم $f(m) = \theta(m^{log_2 2}) = \theta(m)$. بنابراین $S(m) = \theta(mlg(m))$. بنابراین داریم:

$$S(m) = T(2^m) = \theta(mlg(m))$$
$$\Rightarrow T(n) = \theta(lg(n)lg(lg(n)))$$

سوال ۲ - الف)

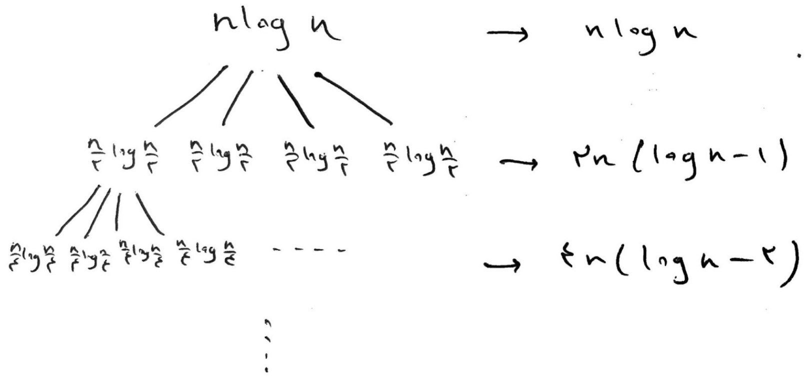
درخت بازگشت $T(n)$ به صورت زیر است:



شاخه سمت چپ درخت، که حاصل $T(\frac{n}{4})$ است، زمانی به به انتها می‌رسد که $\frac{n}{4^i} = 1$ که در آن i همان شماره سطر انتهایی است. بنابراین $i = log_4 n$ و این شاخه تا سطر $log_4 n$ ادامه می‌یابد. به طور مشابه، شاخه سمت راست نیز تا سطر $log_{\frac{4}{3}} n$ ادامه می‌یابد. در نتیجه درخت حداقل $log_4 n$ و حداکثر $log_{\frac{4}{3}} n$ سطر دارد. با توجه به این که هزینه هر سطح برابر n است، در نتیجه $T(n) = O(nlogn)$ و $T(n) = \Omega(nlogn)$. بنابراین $T(n) = \theta(nlogn)$ (البته هزینه هرکدام از سطرهای بین سطر $log_4 n$ و $log_{\frac{4}{3}} n$ کمتر از n است، ولی این موضوع تناقضی با اثبات بالا ندارد).

سوال ۲ - ب)

درخت بازگشت $T(n)$ به صورت زیر است:



در سطر i ام، 4^i گره وجود دارد. همچنین هزینه هر گره در سطح i ام برابر $\frac{n}{2^i}log(\frac{n}{2^i}) = \frac{n}{2^i}(log(n) - i$ است. بنابراین هزینه هر سطح برابر $4^i \times \frac{n}{2^i}(log(n) - i) = 2^i \times n(log(n) - i$ است. همچنین ارتفاع درخت برابر با $lg(n)$ است. بنابراین:

$$T(n) = \sum_{i=0}^{lg(n)} 2^i \times n(log(n) - i)$$

داریم:

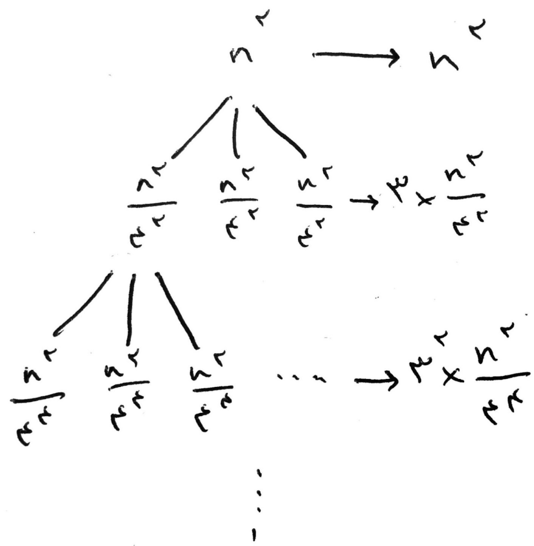
$$T(n) < \sum_{i=0}^{lg(n)} 2^i \times n(log(n)) = n(log(n)) \times \sum_{i=0}^{lg(n)} 2^i = 2 \times 2^{lg(n)} \times n(log(n))$$

$$\Rightarrow T(n) < 2n^2 log(n)$$

بنابراین $T(n) = O(n^2 log(n))$.

سوال ۲ - ج)

درخت بازگشت $T(n)$ به صورت زیر است:



هزینه هر گره در سطر i برابر $(\frac{n}{4^i})^2 = \frac{n^2}{16^i}$ است. همچنین در سطر i ام، 3^i گره داریم. بنابراین مجموع هزینه هر سطر، برابر $(\frac{3}{16})^i n^2$ است. در کل نیز $log_4 n$ سطر داریم. بنابراین مجموع هزینه کل درخت برابر است با $\sum_{i=o}^{log_4 n} (\frac{3}{16})^i n^2$. برای ساده شدن کار، مجموع را تا بی‌نهایت حساب می‌کنیم. داریم:

$$T(n) < \sum_{i=0}^{\infty} (\frac{3}{16})^i n^2 = n^2 \sum_{i=0}^{\infty} (\frac{3}{16})^i = n^2 \times \frac{1}{1-\frac{3}{16}} = \frac{16}{13} n^2$$

بنابراین $T(n) = O(n^2)$.

سوال ۳

```
EUCLIDEAN_GCD(N, M):
  IF (M > N): SWAP(N, M)

  IF (M == 0) RETURN N
  ELSE RETURN EUCLIDEAN_GCD(M, N % M)
```

با توجه به زیربرنامه بالا، با فرض این که $n \geq m$ ، تابع هزینه زمانی به صورت زیر است (منظور از عدد ۱، تابعی از $O(1)$ است):

$T(n,\ m) = T(m,\ n\ \%\ m) + 1$

همچنین در صورتی که $n < m$ ، با یک عمل $O(1)$ می‌توان جای n و m را عوض کرد. در نتیجه تابع $T(n)$ تغییری نمی‌کند.

به ازای هربار اجرا شدن زیر برنامه، $O(1)$ عمل انجام می‌شود. بنابراین کافی است تعداد اجرای زیربرنامه‌ها را بشماریم. با توجه به این که داریم $n \geq m$ ، بنابراین $n + m \geq 2m$. همچنین می‌دانیم $n\%m < m$ ، بنابراین $m + n\%m < 2m$. نتیجه می‌گیریم که مجموع ورودی‌های زیربرنامه، در هربار اجرا حداقل یک واحد کاهش می‌یابد. بنابراین زیربرنامه نهایتا m+n بار اجرا می‌شود. بنابراین $T(n,\ m) = O(n + m)$.

سوال ۴

```
LIMITED_HANOI(N, FROM, USING, TO):
  IF N==1:
    IF (FROM == ‘c’ AND TO == ‘a’) OR (FROM == ‘a’ AND TO == ‘c’):
      MOVE(FROM, USING)
      MOVE(USING, TO)
    ELSE:
      MOVE(FROM, TO)

  LIMITED_HANOI(N-1, FROM, USING, TO)
  MOVE(FROM, USING)
  LIMITED_HANOI(N-1, TO, USING, FROM)
  MOVE(USING, TO)
  LIMITED_HANOI(N-1, FROM, USING, TO)
```

تابع هزینه زمانی این زیربرنامه را $T(n)$ می‌نامیم. با فرض این که مرتبه هر move از $O(1)$ است، $T(n)$ به صورت زیر تعریف می‌شود (در اینجا منظور از عدد ۱، تابعی از $O(1)$ است):

$$T(n) = \begin{cases} 1 & n = 1 \\ 3T(n-1) + 1 & n > 1 \end{cases}$$

با بسط دادن رابطه داریم:

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

$$\Rightarrow T(n) = 3^kT(n-k) + \sum_{i=0}^{k-1} 3^i$$

با توجه به شرط پایه داریم:

$$n - k = 1 \Rightarrow k = n - 1$$

$$\Rightarrow T(n) = 3^{n-1} + \sum_{i=0}^{n-2} 3^i = \sum_{i=0}^{n-1} 3^i = \frac{1(3^n-1)}{3-1} = \frac{1}{2} \times 3^n - \frac{1}{2}$$

بنابراین $T(n) \in O(3^n)$.

سوال ۵ - a)

داریم:

$$a = 27,\ b = 3,\ f(n) = n^3log^2(n)$$

داریم:

$$\frac{n^3log^2(n)}{n^{log_3 27}} = \frac{n^3log^2(n)}{n^3} = log^2(n)$$

بنابراین نمی‌توان هیچ $\varepsilon > 0$ را پیدا کرد که یکی از شرایط قضیه اصلی را برقرار کنند. بنابراین با قضیه اصلی نمی‌توان مرتبه رابطه بازگشتی را حساب کرد.

سوال ۵ - b)

داریم $a = 2^n$، در حالی که a باید عددی ثابت باشد. بنابراین با قضیه اصلی نمی‌توان مرتبه رابطه بازگشتی را حساب کرد.

سوال ۵ - c)

داریم:

$$a = 3,\ b = 2,\ f(n) = n^2$$

به ازای $\varepsilon = 0.1$، داریم $f(n) = \Omega(n^{lg(3)+\varepsilon}) = \Omega(n^{1.68})$. بنابراین $T(n) = \theta(f(n)) = \theta(n^2)$.

سوال ۵ - d)

داریم:

$$a = 1,\ b = 2,\ f(n) = 2n - ncos(n)$$

ابتدا نشان می‌دهیم $f(n) = \Omega(n)$. می‌دانیم $-1 \le cos(n) \le 1$. بنابراین:

$$-n \le -ncos(n) \le n$$

$$\Rightarrow n \le 2n - ncos(n) \le 3n$$

پس $f(n) = \Omega(n)$. حال می‌دانیم به ازای $\varepsilon = 1$، داریم $f(n) = \Omega(n^{lg(1)+\varepsilon}) = \Omega(n)$. بنابراین $T(n) = \theta(n)$.

سوال ۵ - e)

داریم:

$a = 2,\ b = 2,\ f(n) = n^2$

به ازای $\varepsilon = 1$، داریم $f(n) = \Omega(n^{lg(2)+\varepsilon}) = \Omega(n^2)$. بنابراین $T(n) = \theta(f(n)) = \theta(n^2)$.

سوال ۵ - f)

چون $f(n) =- nlog(n)$ در بی‌نهایت مثبت نیست، بنابراین با استفاده از قضیه اصلی نمی‌توان مرتبه این رابطه بازگشتی را حساب کرد.

سوال ۶

فرض استقرا: فرض می‌کنیم $T(1) = 1$. ثابت می‌کنیم حکم برای $T(2)$ درست است:

$T(2) = 2T(1) + 1 = 3 \geq c.log(2)$

$\Rightarrow c \leq \frac{3}{log(2)}$

پس به ازای هر $c \leq \frac{3}{log(2)}$، حکم برای $T(2)$ برقرار است.

گام استقرا: فرض می‌کنیم حکم برای هر مقدار کوچکتر از n، مثلا $\frac{n}{2}$، درست است. یعنی:

$T(\frac{n}{2}) \geq c_1.log(\frac{n}{2})$

حال ثابت می‌کنیم حکم برای n درست است:

$T(n) \geq c_2.log(n)$

با جایگزینی فرض در حکم سوال داریم:

$T(n) = 2T(\frac{n}{2}) + 1 \geq 2c_1.log(\frac{n}{2}) + 1$

$\Rightarrow T(n) \geq 2c_1.log(n) - (2c_1.log(2) - 1)$

برای این که حکم درست باشد، کافی است $2c_1.log(2) - 1 \geq 0$:

$2c_1.log(2) - 1 \geq 0 \Rightarrow c_1 \geq \frac{1}{2log(2)}$

پس کافی است $c_2 = 2c_1 \geq \frac{1}{log(2)}$ باشد تا حکم درست باشد. بنابراین حکم درست است.
