CS170 HW01

Jialiang Tang

tangjialiang@berkeley.edu

8/30/2023

1 Study Group

Jialiang Tang(myself), SID:3039758308

Buji Xu, SID:3039687809

2 Course Policies

(a)
Midterm1: 10.3 19:00-21:00
Midterm2: 11.7 19:00-21:00
Final: 12.15 8:00-11:00
There is no planned alternate exam.

(b)
Before Mondays at 10 pm.

(c)
Nothing.

(d)
Ed discussion.

(e)
I have read and understood the course syllabus and policies.

3 Understanding Academic Dishonesty

(a)
Not OK.

(b)
Not OK.

(c)
Not OK.

(d)
OK.

4 Math Potpourri

(a)

(i) $\frac{\ln x}{\ln y} = \log_y x$

(ii) $\ln x + \ln y = \ln xy$

(iii) $\ln x$ - $\ln y = \ln \frac{x}{y}$

(iv) $170 \ln x = \ln x^{170}$

(b)

(a)

Proof: $x^{\log_{1/x} y} = x^{\frac{\ln y}{\ln 1/x}} = x^{\frac{\ln y}{-\ln x}} = x^{\frac{-\ln y}{\ln x}} = x^{\frac{\ln 1/y}{\ln x}} = x^{\log_x 1/y} = \frac{1}{y}$

(b)

Proof: $\sum_{i=1}^{n} i = \frac{1}{2} \cdot 2 \sum_{i=1}^{n} i = \frac{1}{2}(\sum_{i=1}^{n} i + \sum_{i=1}^{n}(n+1-i)) = \frac{1}{2}(\sum_{i=1}^{n}(n+1))$ $= \frac{1}{2}n(n+1)$

(c)

Proof: If r = 1, then $\sum_{k=0}^{n} ar^k = \sum_{k=0}^{n} a = a(n+1)$. Else, we multiply $\sum_{k=0}^{n} ar^k$ by r, then we get $\sum_{k=1}^{n+1} ar^k$. Make a subtraction between the two equations, we have

$$\sum_{k=1}^{n+1} ar^k - \sum_{k=0}^{n} ar^k = (r-1) \sum_{k=0}^{n} ar^k = ar^{n+1} - a$$

Since $r \neq 1$, we have

$$\sum_{k=0}^{n} ar^k = a \left(\frac{1 - r^{n+1}}{1 - r} \right)$$

To sum, it holds that

$$\sum_{k=0}^{n} ar^k = \begin{cases} a\left(\frac{1-r^{n+1}}{1-r}\right), & r \neq 1 \\ a(n+1), & r = 1 \end{cases}$$

5 Recurrence Relations

(a) Using the recurrence relation, we have

$$\begin{aligned}
T(n) &= 3T(n/4) + 10n \\
&= 3(3T(n/4^2) + 10 \cdot n/4) + 10n \\
&= \dots \\
&= 3^{\log_4 n} T(1) + 10n[1 + (3/4) + (3/4)^2 + ... + (3/4)^{\log_4 n - 1}] \\
&= 3^{\log_4 n} + 10n \cdot \frac{1 - (3/4)^{\log_4 n}}{1 - (3/4)} \\
&= 3^{\log_4 n} + 40n - 40 \cdot 3^{\log_4 n} \\
&= 40n - 39 \cdot n^{log_4 3}
\end{aligned}$$

Since

$$\lim_{n \to \infty} \frac{T(n)}{n} = \lim_{n \to \infty} \frac{40n - 39 \cdot n^{log_4 3}}{n} = \lim_{n \to \infty} 40 - 39 n^{\log_4 3 - 1} = 40$$

So $T(n) = \Theta(n)$.

(b) By the definition of big-Θ, we have

$$\begin{aligned}
T(n) &= 97T(n/100) + \Theta(n) \\
&= 97(97T(n/100^2) + \Theta(n/100) + \Theta(n) \\
&= \dots \\
&= 97^{\log_{100} n} T(1) + \Theta(n(1 + 97/100 + (97/100)^2 + \cdots + (97/100)^{\log_{100} n - 1})) \\
&= 97^{\log_{100} n} + \Theta(\frac{1 - (97/100)^{\log_{100} n}}{1 - (97/100)} \cdot n) \\
&= n^{\log_{100} 97} + \Theta(n - n^{\log_{100} 97}) \\
&= \Theta(n^{\log_{100} 97}) + \Theta(n - n^{\log_{100} 97}) \\
&= \Theta(n)
\end{aligned}$$

(c) The recurrence relation is

$$T(n) = 3T(n/5) + \Theta(n^2)$$

Then we have

$$\begin{aligned}
T(n) &= 3T(n/5) + \Theta(n^2) \\
&= 3(3T(n/5^2) + \Theta((n/5)^2) + \Theta(n^2) \\
&= \dots \\
&= 3^{\log_5 n}T(1) + \Theta(n^2(1 + 3/5^2 + (3/5^2)^2 + \cdots + (3/5^2)^{\log_5 n-1})) \\
&= 3^{\log_5 n} + \Theta(\frac{1-(3/5^2)^{\log_5 n}}{1-(3/5^2)} \cdot n^2) \\
&= n^{\log_5 3} + \Theta(n^2 - n^{\log_5 3}) \\
&= \Theta(n^{\log_5 3}) + \Theta(n^2 - n^{\log_5 3}) \\
&= \Theta(n^2)
\end{aligned}$$

(d) By the definition of big-Θ, we have

$$\begin{aligned}
T(n) &= 3T(n/3) + \Theta(n) \\
&= 3(3T(n/3^2) + \Theta(n/3) + \Theta(n) \\
&= \dots \\
&= 3^{\log_3 n}T(1) + \Theta(n(1 + 1 + 1^2 + \cdots + 1^{\log_3 n-1})) \\
&= n + \Theta(\log_3 n \cdot n)
\end{aligned}$$

Since

$$\lim_{n\to\infty} \frac{T(n)}{n\log_3 n} = \lim_{n\to\infty} \frac{n + \Theta(n\log_3 n)}{n\log_3 n} = C$$

where C is constant and $C > 0$ We have

$$T(n) = \Theta(n\log_3 n)$$

(e) Let's guess $T(n) = n^2$. Then we use Second Mathematical Induction to prove the equation.
First, we can see $T(1) = 1 = 1^2$, so it holds when $n = 1$.
Next, we assume that it holds for all $n \leq k$, where k is a positive integer. That's to say, $T(n) = n^2$ for all $n \leq k$. By the recurrence relation we have$T(k+1) = T(3(k+1)/5)+T(4(k+1)/5) = (3(k+1)/5)^2+(4(k+1)/5)^2 = ((k+1)/5)^2$.
So it also holds when $n = k+1$. By second mathematical induction, we have

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

for all $n \in \mathbb{N}^*$. Obviously $T(n) = \Theta(n^2)$.

6 In Between Functions

(a) For $k > d$, f fail to satisfy (1).
Proof: We can assume that $a_d > 0$.
Firstly, for $k \le d$, we have

$$\lim_{n\to\infty} \frac{f(n)}{n^k} = \lim_{n\to\infty} \frac{a_0 + a_1 n + \cdot + a_d n^d}{n^k} > \lim_{n\to\infty} a_d n^{d-k} > 0$$

So $f(n) = \Omega(n^k)$, which satisfies (1).
Then, for $k > d$, we have

$$\lim_{n\to\infty} \frac{f(n)}{n^k} = \lim_{n\to\infty} \frac{a_0 + a_1 n + \cdot + a_d n^d}{n^k} = \lim_{n\to\infty} \sum_{i=0}^{d} a_i n^{i-k} = \sum_{i=0}^{d} \lim_{n\to\infty} a_i n^{i-k} = \sum_{i=0}^{d} 0 = 0$$

To sum, for $k > d$, f fail to satisfy (1). $\square$

(b) It's easy to see $a > 1$ since a should make f satisfy (1).
For $c < \log_2 a$, f fail to satisfy (2).
Proof: f satisfies (2) is equivalent to

$$\lim_{n\to\infty} \frac{a^n}{2^{cn}} < \infty$$

which is equivalent to

$$\lim_{n\to\infty} (\frac{a}{2^c})^n < \infty$$

It's a exponential function, we have $\frac{a}{2^c} \le 1$, which means $c \ge \log_2 a$, so $c < \log_2 a$. $\square$

(c) We assign $D(n) = \sqrt{n}$. Then we claim that both (1) and (2) hold.
Proof: For (1), we use the limitation

$$\lim_{n\to\infty} \frac{f(n)}{n^k} = \lim_{n\to\infty} \frac{n^{\sqrt{n}}}{n^k} = \lim_{n\to\infty} n^{\sqrt{n}-k} = \infty > 0$$

So (1) holds.
For (2), we have

$$\lim_{n\to\infty} \frac{n^{\sqrt{n}}}{2^{cn}} = \lim_{n\to\infty} e^{\sqrt{n}\ln n - cn\ln 2} = e^{\lim_{n\to\infty}(\sqrt{n}\ln n - cn\ln 2)} = e^{-\infty} = 0 < \infty$$

So (2) holds. $\square$
Note: Actually, we only need to make $D(n)$ "greater" than 1 and "smaller" than $\frac{n}{\log n}$.

7 Decimal to Binary

(a) Algorithm Description: Divide the n-digit decimal number to the first $n/2$ digits(noted as n_1) and the last $n/2$ digits(noted as n_2).
Then $n = 10^{n/2}n_1 + n_2$, which means we only need to make twice transfers on $n/2$-digit n_1, n_2 and $10^{n/2}$, and do a at most $\lceil \frac{n}{2} \cdot \log_2 10 \rceil$ binary multiplication(because a n-digit decimal number has at most $\lceil n \log_2 10 \rceil$ digits in its binary representation) and at most $2\lceil \frac{n}{2} \cdot \log_2 10 \rceil$-digit binary addition.

(b) Assume the time complexity of algorithm in (a) is T(n) where n is the digit number of a decimal number. The recurrence relationship is

$$T(n) = 3T(n/2) + \mathcal{O}(n^{\log_2 3})$$

Since we handle 3 sub-problems whose scale is $n/2$. The time complexity of the binary multiplication and addition is $\Theta(n^{\log_2 3})$ using Karatsuba's algorith. Because the digits of binary numbers to multiply are less than $\lceil \frac{n}{2} \cdot \log_2 10 \rceil < 5n$, on the other hand, the complexity of binary addition is $\Theta(n)$, so the time complexity is at most $\mathcal{O}((5n)^{\log_2 3} + n) = \mathcal{O}(n^{log_2 3})$.
Assume that $T(1) = 1$, we have

$$\begin{aligned}
T(n) &= 3T(n/2) + \mathcal{O}(n^{\log_2 3}) \\
&= 3(3T(n/2^2) + \mathcal{O}((n/2)^{\log_2 3}) + \mathcal{O}(n^{\log_2 3}) \\
&= \dots \\
&= 3^{\log_2 n}T(1) + \mathcal{O}(n^{\log_2 3}(1 + 3(1/2)^{\log_2 3} + \cdot + 3^{\log_2 n-1}(1/2^{\log_2 n-1})^{\log_2 3})) \\
&= 3^{\log_2 n} + \mathcal{O}(n^{\log_2 3}(1 + 1^1 + \cdot + 1^{\log_2 n-1})) \\
&= n^{\log_2 3} + \mathcal{O}(n^{\log_2 3}\log_2 n) \\
&= \mathcal{O}(n^{\log_2 3}\log n)
\end{aligned}$$

So the desired running time is $\mathcal{O}(n^{\log_2 3}\log n)$.