CS 5381 Analysis of Algorithms
Solutions to Homework 1

Fall 2022

1. Define the function $h(n) = \max(f(n), g(n))$, so that

$$h(n) = \begin{cases} f(n) & \text{if } f(n) \geq g(n) \\ g(n) & \text{if } f(n) < g(n). \end{cases}$$

Since $f(n)$ and $g(n)$ are asymptotically nonnegative, there exists a n_0 such that $f(n) \geq 0$ and $g(n) \geq 0$ for all $n \geq n_0$.

Thus, for $n \geq n_0$, $f(n) + g(n) \geq f(n) \geq 0$ and $f(n) + g(n) \geq g(n) \geq 0$. Since for any particular n, $h(n)$ is either $f(n)$ or $g(n)$, we have $f(n) + g(n) \geq h(n) \geq 0$. This shows that $h(n) = \max(f(n), g(n)) \leq c_2(f(n) + g(n))$ for all $n \geq n_0$, where $c_2 = 1$.

Similarly, we have for all $n \geq n_0$, $0 \leq f(n) \leq h(n)$ and $0 \leq g(n) \leq h(n)$. Adding these two inequalities yields $0 \leq (f(n) + g(n))/2 \leq h(n)$. This shows that $h(n) = \max(f(n), g(n)) \geq c_1(f(n) + g(n))$ for all $n \geq n_0$, where $c_1 = 1/2$.

2. We need to find constants c_1, c_2, $n_0 > 0$ such that

$$0 \leq c_1 n^b \leq (n + a)^b \leq c_2 n^b, \quad \forall n \geq n_0.$$

Observe that when $|a| \leq n$, one has

$$n + a \leq n + |a| \leq 2n$$

and when $|a| \leq n/2$, one has

$$n + a \geq n - |a| \geq n/2.$$

Thus, when $n \geq 2|a|$, it holds

$$0 \leq n/2 \leq n + a \leq 2n.$$

Since $b > 0$, the inequality still holds when all parts are raised to the power of b as

$$0 \leq (1/2)^b n^b \leq (n + a)^b \leq 2^b n^b.$$

Thus, we obtain $c_1 = (1/2)^b$, $c_2 = 2^b$, and $n_0 = 2|a|$.

3. We conjecture that $T(n) \leq cn^2$ for some constant $c > 0$. We have

$$\begin{aligned} T(n) &= T(n-1) + n \\ &\leq c(n-1)^2 + n \\ &= cn^2 + c(1 - 2n) + n. \end{aligned}$$

The last quantity is less than or equal to cn^2 if $c(1 - 2n) + n \leq 0$ or, equivalently, $c \geq n/(2n - 1)$, which holds for all $n \geq 1$ and $c \geq 1$. Thus, we can choose $n_0 = 1$ and $c = 1$.

4. Using the recursion tree shown in Figure 1 on the next page, we get a guess of $T(n) = \Theta(n)$.

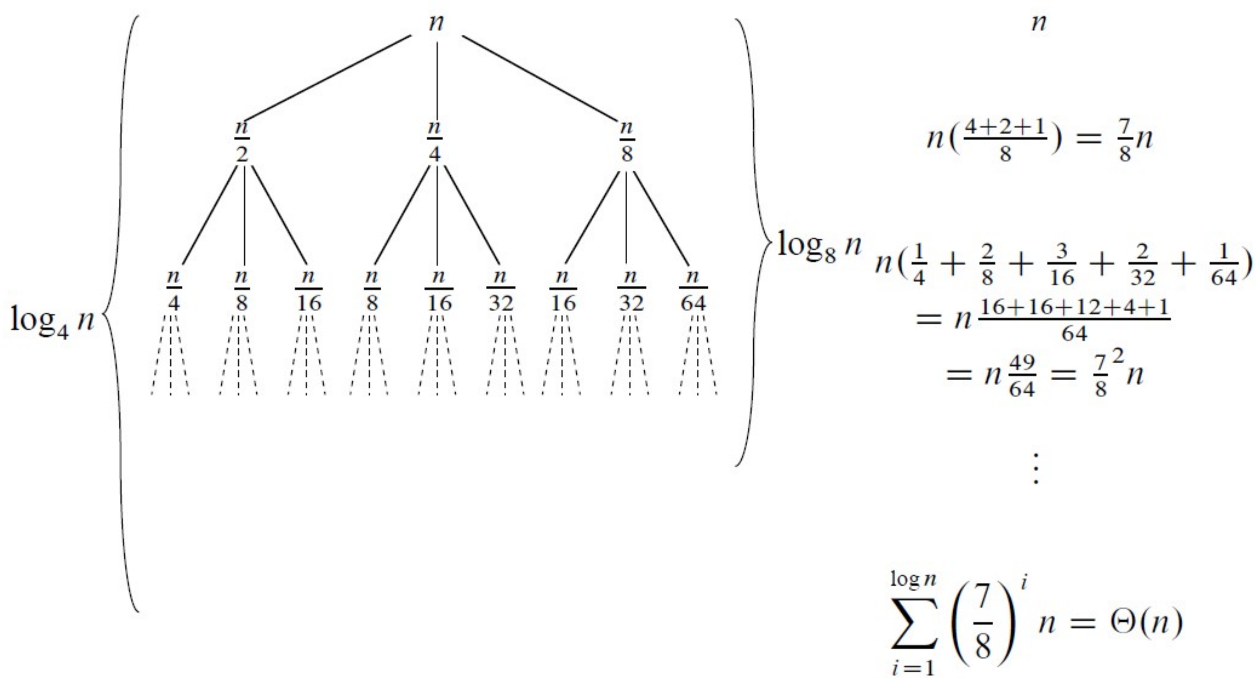

Figure 1: Problem 4

We then use the substitution method to first prove that $T(n) = O(n)$. Our inductive hypothesis is that $T(n) \leq cn$ for some constant $c > 0$. We have

$$\begin{aligned} T(n) &= T(n/2) + T(n/4) + T(n/8) + n \\ &\leq cn/2 + cn/4 + cn/8 + n \\ &= (1 + 7c/8)n \\ &\leq cn \quad \text{if } c \geq 8. \end{aligned}$$

Thus, $T(n) = O(n)$. Showing that $T(n) = \Omega(n)$ is easy:

$$T(n) = T(n/2) + T(n/4) + T(n/8) + n \geq n.$$

Therefore, we have $T(n) = \Theta(n)$.

5. We have $a = 8$, $b = 2$, and $f(n) = \Theta(n^2)$, and so $n^{\log_b a} = n^{\log_2 8} = n^3$. Since n^3 is polynomially larger than $f(n)$, case 1 of the master theorem applies, and $T(n) = \Theta(n^3)$.

6. We have $a = 2$, $b = 2$, and $f(n) = n^3$, and so $n^{\log_b a} = n^{\log_2 2} = n$. Since $f(n)$ is polynomially larger than $n^{\log_b a}$, case 3 of the master theorem applies, and $T(n) = \Theta(n^3)$, where the regularity condition can also be easily verified.

7. We have $a = 3$, $b = 2$, and $f(n) = n\mathrm{lg}n$, and so $n^{\log_b a} = n^{\mathrm{lg}3} \approx n^{1.58}$. Since $n\mathrm{lg}n = O(n^{\mathrm{lg}3-\epsilon})$ for any $0 < \epsilon \leq 0.58$, case 1 of the master theorem applies, and $T(n) = \Theta(n^{\mathrm{lg}3})$.

8. Since HIRE-ASSISTANT always hires candidate 1, it hires exactly once if and only if no candidates other than candidate 1 are hired. This event occurs when candidate 1 is the best candidate of the n, which occurs with probability $1/n$.

HIRE-ASSISTANT hires n times if each candidate is better than all those who were interviewed (and hired) before. This event occurs precisely when the candidates come in strictly increasing order of quality, which occurs with probability $1/n!$.

9. Define a random variable X that equals the number of customers that get back their own hat, so that we want to compute $E[X]$. For $i = 1, 2, \ldots, n$, define the indicator random variable

$$X_i = I\{\text{customer } i \text{ gets back his or her own hat}\}.$$

Then $X = X_1 + X_2 + \cdots + X_n$.

Since the ordering of hats is random, each customer has a probability of $1/n$ of getting back his or her own hat. That is $E[X] = Pr\{X_i = 1\} = 1/n$. Therefore,

$$\begin{aligned} E[X] &= E\left[\sum_{i}^{n} X_i\right] \\ &= \sum_{i=1}^{n} E[X_i] \\ &= \sum_{i=1}^{n} \frac{1}{n} = 1, \end{aligned}$$

and so we expect that exactly 1 customer gets back his or her own hat.