CSCI2100: Regular Exercise Set 3

Prepared by Yufei Tao

Problem 1. Prove $\log_2(n!) = \Theta(n \log n)$.

Solution. Let us prove first $\log_2(n!) = O(n \log n)$:

$$\begin{aligned} \log_2(n!) &= \log_2(\Pi_{i=1}^{n} i) \\ &\le \log_2 n^n \\ &= n \log_2 n \\ &= O(n \log n). \end{aligned}$$

Now we prove $\log_2(n!) = \Omega(n \log n)$:

$$\begin{aligned} \log_2(n!) &= \log_2(\Pi_{i=1}^{n} i) \\ &\ge \log_2(\Pi_{i=n/2}^{n} i) \\ &\ge \log_2(n/2)^{n/2} \\ &= (n/2)\log_2(n/2) \\ &= \Omega(n \log n). \end{aligned}$$

This completes the proof.

Problem 2. Let $f(n)$ be a function of positive integer n. We know:

$$\begin{aligned} f(1) &= 1 \\ f(n) &\le 2 + f(\lceil n/10 \rceil). \end{aligned}$$

Prove $f(n) = O(\log n)$. Recall that $\lceil x \rceil$ is the ceiling operator that returns the smallest integer at least x.

Solution 1 (Expansion). Consider first n being a power of 10.

$$\begin{aligned} f(n) &\le 2 + f(n/10) \\ &\le 2 + 2 + f(n/10^2) \\ &\le 2 + 2 + 2 + f(n/10^3) \\ &\ldots \\ &\le 2 \cdot \log_{10} n + f(1) \\ &= 2 \cdot \log_{10} n + 1 = O(\log n). \end{aligned}$$

Now consider n that is not a power of 10. Let n' be the smallest power of 10 that is greater than n. We have:

$$\begin{aligned} f(n) &\le f(n') \\ &\le 2\log_{10} n' + 1 \\ &\le 2\log_{10}(10n) + 1 \\ &\le O(\log n). \end{aligned}$$

Solution 2 (Master Theorem). Let α, β, γ, and λ be as defined in the Master Theorem. Thus, we have $\alpha = 1, \beta = 10$, $\gamma = 0$, and $\lambda = 0$. Since $\log_\beta \alpha = \log_{10} 1 = 0 = \gamma$, by the Master Theorem, we know that $f(n) = O(n^\gamma \log^{\lambda+1} n) = O(\log n)$.

Solution 3 (Substitution). We aim to find constants α, β such that $f(n) \le \alpha \log_2 n$ for all $n \ge \beta$.

Base case: $n = \beta$. We need

$$f(\beta) \le \alpha \log_2 \beta \quad \Leftrightarrow \quad \alpha \ge f(\beta)/\log_2 \beta. \tag{1}$$

Inductive case. Assuming $f(n) \le \alpha \log_2 n$ for all $n \le t-1$ where $t \ge \beta + 1$, we want to prove $f(t) \le \alpha \log_2 t$.

We will consider only

$$\beta \quad \ge \quad 2 \tag{2}$$

such that $t \ge 3$ and, hence, $\lceil t/10 \rceil \le (t/10) + 1 \le t/2$. With this, we have:

$$\begin{aligned} f(t) &\le 2 + f(\lceil t/10 \rceil) \\ &\le 2 + \alpha \log_2 \lceil t/10 \rceil \\ &\le 2 + \alpha \log_2 (t/2) \\ &= 2 + \alpha \log_2 t - \alpha. \end{aligned}$$

To complete the inductive argument, we need the above to be at most $\alpha \log_2 t$, namely:

$$\alpha \ge 2. \tag{3}$$

To satisfy (1)-(3), we set $\beta = 2$ and $\alpha = \max\{2, f(2)/\log_2 2\}$.

Problem 3. Let $f(n)$ be a function of positive integer n. We know:

$$\begin{aligned} f(1) &= 1 \\ f(n) &\le 2n + 4f(\lceil n/4 \rceil). \end{aligned}$$

Prove $f(n) = O(n \log n)$.

Solution 1 (Expansion). Consider first n being a power of 4.

$$\begin{aligned} f(n) &\le 2n + 4f(n/4) \\ &\le 2n + 4(2n/4 + 4f(n/4^2)) \\ &\le 2n + 2n + 4^2 f(n/4^2) \\ &= 2 \cdot 2n + 4^2 f(n/4^2) \\ &\le 2 \cdot 2n + 4^2 \cdot (2(n/4^2) + 4f(n/4^3)) \\ &= 3 \cdot 2n + 4^3 f(n/4^3) \\ &\dots \\ &= (\log_4 n) \cdot 2n + n \cdot f(1) \\ &= (\log_4 n) \cdot 2n + n = O(n \log n). \end{aligned}$$

Now consider that n is not a power of 4. Let n' be the smallest power of 4 that is greater than n. This implies that $n' < 4n$. We have:

$$\begin{aligned} f(n) &\leq f(n') \\ &\leq (\log_4 n') \cdot 2n' + n' \\ &< (\log_4(4n)) \cdot 8n + 4n = O(n \log n). \end{aligned}$$

Solution 2 (Master Theorem). Let α, β, γ, and λ be as defined in the Master Theorem. Thus, we have $\alpha = 4, \beta = 4$, $\gamma = 1$, and $\lambda = 0$. Since $\log_\beta \alpha = \log_4 4 = 1 = \gamma$, by the Master Theorem, we know that $f(n) = O(n^\gamma \log^{\lambda+1} n) = O(n \log n)$.

Solution 3 (Substitution). We aim to find constants α, β such that $f(n) \leq \alpha n \log_2 n$ for all $n \geq \beta$.

Base case: $n = \beta$. We need:

$$f(\beta) \leq \alpha \cdot \beta \log_2 \beta \quad \Leftrightarrow \quad \alpha \geq f(\beta)/(\beta \log_2 \beta). \tag{4}$$

Inductive case. Assuming $f(n) \leq \alpha n \log_2 n$ for all $n \leq t - 1$ where $t \geq \beta + 1$, we want to prove $f(t) \leq \alpha t \log_2 t$.

We will consider only

$$\beta \quad \geq \quad 4 \tag{5}$$

such that $t \geq 5$ and, hence, $t/4 + 1 \leq t/2$. With this, we have:

$$\begin{aligned} f(t) &\leq 2t + 4\alpha \lceil t/4 \rceil \log_2 \lceil t/4 \rceil \\ &\leq 2t + 4\alpha(t/4 + 1) \log_2(t/4 + 1) \\ &\leq 2t + 4\alpha(t/4 + 1) \log_2(t/2) \\ &= 2t + (\alpha t + 4\alpha)(\log_2 t - 1) \\ &\leq 2t + (\alpha t + 4\alpha) \log_2 t - \alpha t - 4\alpha \\ &\leq 2t + \alpha t \log_2 t + 4\alpha \log_2 t - \alpha t - 4\alpha \end{aligned}$$

To complete the inductive argument, we need the above to be at most $\alpha t \log_2 t$, namely:

$$2t + 4\alpha \log_2 t \quad \leq \quad \alpha t + 4\alpha \tag{6}$$

We will make sure

$$\beta \quad \geq \quad 2^8. \tag{7}$$

Under the above condition, for any $t \geq \beta$, it holds that $\log_2 t \leq t/8$. To ensure (6), it suffices to have:

$$\begin{aligned} 2t + 4\alpha(t/8) &\leq \alpha t + 4\alpha \\ \Leftrightarrow 2t + \alpha t/2 &\leq \alpha t + 4\alpha \\ \Leftrightarrow 2t &\leq \alpha t/2 + 4\alpha \end{aligned}$$

which always holds when

$$\alpha \geq 4. \tag{8}$$

To satisfy (4), (5), (7), and (8), we set $\beta = 2^8$ and $\alpha = \max\{f(2^8)/(2^8 \log 2^8), 5\}$.

Problem 4 (Bubble Sort). Let us re-visit the sorting problem. Recall that, in this problem, we are given an array A of n integers, and need to re-arrange them in ascending order. Consider the following *bubble sort* algorithm:

1. If $n = 1$, nothing to sort; return.
2. Otherwise, do the following in ascending order of $i \in [1, n-1]$: if $A[i] > A[i+1]$, swap the integers in $A[i]$ and $A[i+1]$.
3. Recurse in the part of the array from $A[1]$ to $A[n-1]$.

Prove that the algorithm terminates in $O(n^2)$ time.

As an example, support that A contains the sequence of integers $(10, 15, 8, 29, 13)$. After Step 2 has been executed once, array A becomes $(10, 8, 15, 13, 29)$.

Solution. Let $f(n)$ be the worst-case running time of bubble sort when A has n elements. From Step 1, we know:

$$f(1) = O(1).$$

From Steps 2-3, we know:

$$f(n) \leq f(n-1) + O(n).$$

Solving the recurrence (e.g., by the expansion method) gives $f(n) = O(n^2)$.

Problem 5* (Modified Merge Sort). Let us consider a variant of the merge sort algorithm for sorting an array A of n elements (we will use the notation $A[i..j]$ to represent the part of the array from $A[i]$ to $A[j]$):

- If $n = 1$ then return immediately.
- Otherwise set $k = \lceil n/3 \rceil$.
- Recursively sort $A[1..k]$ and $A[k+1..n]$, respectively.
- Merge $A[1..k]$ and $A[k+1..n]$ into one sorted array.

Prove that this algorithm runs in $O(n \log n)$ time.

Solution. Let $f(n)$ be the worst case time of the algorithm on an array of size n. We have the following recurrence:

$$\begin{aligned} f(1) &= O(1) \\ f(n) &\leq f(\lceil n/3 \rceil) + f(\lceil 2n/3 \rceil) + c \cdot n \end{aligned}$$

where $c > 0$ is a constant.

We want to find α, β such that $f(n) \leq \alpha \cdot n \log_2 n$ for all $n \geq \beta$.

Base case: $n = \beta$. We need

$$f(\beta) \le \alpha \cdot \beta \log_2 \beta \quad \Leftrightarrow \quad \alpha \ge f(\beta)/(\beta \log_2 \beta). \tag{9}$$

Inductive case. Assuming $f(n) \le \alpha \cdot n \log_2 n$ for all $n \le t-1$ where $t \ge \beta + 1 \ge 2$, we want to prove $f(t) \le \alpha \cdot t \log_2 t$.

From the recurrence, we get:

$$\begin{aligned} f(t) &\le f(\lceil t/3 \rceil) + f(\lceil 2t/3 \rceil) + c \cdot t \\ &\le \alpha \lceil t/3 \rceil \log_2 \lceil t/3 \rceil + \alpha \lceil 2t/3 \rceil \log_2 \lceil 2t/3 \rceil + c \cdot t \end{aligned}$$

For all $t \ge 2$, we have $\lceil t/3 \rceil \le t/2$ and $\lceil 2t/3 \rceil \le t$. Hence:

$$\begin{aligned} f(t) &\le \alpha(t/3+1)\log_2(t/2) + \alpha(2t/3+1)\log_2 t + ct \\ &= \alpha(t/3+1)((\log_2 t) - 1) + \alpha(2t/3+1)\log_2 t + ct \\ &= \alpha t \log_2 t - t(\alpha/3 - c) - \alpha + 2\alpha \log_2 t \end{aligned}$$

To complete the inductive argument, we want:

$$\begin{aligned} \alpha t \log_2 t - t(\alpha/3 - c) - \alpha + 2\alpha \log_2 t &\le \alpha t \log_2 t \\ \Leftrightarrow \alpha(t/3 - 2\log_2 t + 1) &\ge ct \end{aligned} \tag{10}$$

We consider

$$\beta \ \ge \ 128 \tag{11}$$

under which $t \ge \beta + 1 \ge 129$ and, hence, $t/6 > 2\log_2 t$. Equipped with this, we get from (10):

$$\begin{aligned} \alpha &\ge \frac{ct}{t/3 - 2\log_2 t + 1} \\ \Leftarrow \alpha &\ge \frac{ct}{t/3 - 2\log_2 t} \\ \Leftarrow \alpha &\ge \frac{ct}{t/3 - t/6} \\ &= 6c. \end{aligned} \tag{12}$$

To satisfy (11), (9), and (12), we can choose $\beta = 128$ and $\alpha = \max\{f(128)/(128 \log_2 128), 6c\}$.