COMP3506/7505: Regular Exercise Set 3

Prepared by Yufei Tao

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

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

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

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

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

This completes the proof.

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

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

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

If necessary, you can use without a proof the fact that $f(n)$ is *monotone*, namely, $f(n_1) \leq f(n_2)$ for any $n_1 < n_2$.

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

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

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:

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

Solution 2 (Master Theorem). Let α, β, and γ be as defined in the Master Theorem (see the tutorial slides of Week 4). Thus, we have $\alpha = 1, \beta = 10$, and $\gamma = 0$. Since $\log_\beta \alpha = \log_{10} 1 = 0 = \gamma$, by the Master Theorem, we know that $f(n) = O(n^\gamma \log n) = O(\log n)$.

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

$$\begin{aligned} f(1) &= 1 \\ f(n) &= 2 + f(\lceil 3n/10 \rceil). \end{aligned}$$

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).

$$\begin{aligned} f(n) &= 2 + f(n_1) \quad (\text{define } n_1 = \lceil (3/10)n \rceil) \\ f(n) &= 2 + 2 + f(n_2) \quad (\text{define } n_2 = \lceil (3/10)n_1 \rceil) \\ f(n) &= 2 + 2 + 2 + f(n_3) \quad (\text{define } n_3 = \lceil (3/10)n_2 \rceil) \\ &\ldots \\ f(n) &= \underbrace{2 + 2 + \ldots + 2}_{h \text{ terms}} + f(n_h) \quad (\text{define } n_h = \lceil (3/10)n_{h-1} \rceil) \\ &= 2h + f(n_h). \end{aligned} \tag{1}$$

So it remains to analyze the value of h that makes n_h small enough. Note that we do *not* need to solve the precise value of h; it suffices to prove an upper bound for h. For this purpose, we reason as follows. First, notice that

$$\lceil 3n/10 \rceil \leq (4n/10) \tag{2}$$

when $n \geq 10$ (prove this yourself).

Let us set h to be the smallest integer such that $n_h < 10$ (this implies that $n_{h-1} \geq 10$ and $n_h \geq (4/10)n_{h-1} \geq 4$). We have:

$$\begin{aligned} n_1 &\leq (4/10)n \\ n_2 &= \lceil (3/10)n_1 \rceil \leq (4/10)n_1 \leq (4/10)^2 n \\ n_3 &\leq (4/10)^3 n \\ &\ldots \\ n_h &\leq (4/10)^h n \end{aligned}$$

Therefore, the value of h cannot exceed $\log_{\frac{10}{4}} n$ (otherwise, $(4/10)^4 \cdot n < 1$, making $n_h < 1$, which contradicts the fact that $n_h \geq 4$). Plugging this into (1) gives:

$$f(n) \leq 2\log_{\frac{10}{4}} n + f(10) = O(\log n). \quad (\text{think: why?})$$

Solution 2 (Master Theorem). Let α, β, and γ be as defined in the Master Theorem. Thus, we have $\alpha = 1, \beta = 10/3$, and $\gamma = 0$. Since $\log_\beta \alpha = \log_{10/3} 1 = 0 = \gamma$, by the Master Theorem, we know that $f(n) = O(n^\gamma \log n) = O(\log n)$.

Problem 4. Let $f(n)$ be a function of positive integer n. We know:

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

Prove $f(n) = O(n \log n)$. If necessary, you can use without a proof the fact that $f(n)$ is monotone.

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

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

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, and γ be as defined in the Master Theorem. Thus, we have $\alpha = 4, \beta = 4$, and $\gamma = 1$. Since $\log_\beta \alpha = \log_4 4 = 1 = \gamma$, by the Master Theorem, we know that $f(n) = O(n^\gamma \log n) = O(n \log n)$.

Problem 5 (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. Recur 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 1. Notice that Step 2 is executed $n-1$ times in total. At its j-th ($1 \leq j \leq n-1$) execution, it incurs at most $c \cdot j$ time for some constant $c > 0$. Hence, its worst-case time is no more than

$$c \sum_{j=1}^{n-1} j \quad = \quad cn(n-1)/2 < cn^2 = O(n^2).$$

Solution 2. Let $f(n)$ be the worst-case running time of bubble sort when the array has n elements. From the base case (Step 1), we know:

$$f(1) \quad \leq \quad c_1$$

for some constant c_1. From the inductive case (Steps 2-3), we know:

$$f(n) \quad \leq \quad c_2 n + f(n-1)$$

for some constant c_2. Solving the recurrence (by the expansion method) gives $f(n) = O(n^2)$.