CS344 - Solution to Homework 1

October 19, 2017

Problem 1 (10pt)

Let's first simplify each function:

$$\frac{1}{10} n \log n = \Theta(n \log n)$$

$$\log(n!) = \Theta(n \log n)$$

$$n^2 + \log n = \Theta(n^2)$$

It is obvious that $n \log n = o(n^2)$.
Let's consider $\sqrt{n^{\log n}}$ as follows. Let $x = \sqrt{n^{\log n}}$, taking log for both sides:

$$\log x = \log n^{\frac{\log n}{2}} = \frac{1}{2} \log^2 n \;\Rightarrow\; x = 2^{\frac{\log^2 n}{2}} \approx 1.414^{\log^2 n} = o(2^{\log^2 n})$$

Let's compare $2^{\log^2 n}$ and 2^n: since $\log^2 n = o(n)$, $2^{\log^2 n} = o(2^n)$.
So the final order is

$$\frac{1}{10} n \log n = \Theta(\log n!) = o(n^2 + \log n) = o(2^{\log^2 n}) = o(\sqrt{n^{\log n}}) = o(2^n)$$

Problem 2 (10pt)

No. $2^{2n} = \omega(2^n)$. This is because:

$$\lim_{n \to +\infty} \frac{2^{2n}}{2^n} = \lim_{n \to +\infty} \frac{2^{n+n}}{2^n} = \lim_{n \to +\infty} \frac{2^n * 2^n}{2^n} = \lim_{n \to +\infty} 2^n = \infty$$

which means 2^{2n} dominates 2^n.

Problem 3 (10pt)

$$T(n) = \begin{cases} 1, & \text{if } n = 1 \\ T(\sqrt{n}) + n, & \text{if } n \geq 2 \end{cases}$$

Option 1: Master method Let $n = 2^m$, then for $m \geq 1$, $T(2^m) = T(2^{\frac{m}{2}}) + 2^m$. Let $S(m) = T(2^m)$, then $S(m) = S(m/2) + 2^m$. Considering $S(m)$ through master theorem: $f(m) = 2^m$, $\log_b a = \log_2 1 = 0$, so $2^m = \Omega(n^{0+\epsilon})$ for some constant $\epsilon > 0$. We conclude $S(m) = \Theta(2^m)$, which implies $T(2^m) = \Theta(2^m) \;\Rightarrow\; T(n) = \Theta(n), (n \geq 1)$.

Option 2: Guess & verify We guess $T(n) = \Theta(n)$.
- *Upper bound.* Claim: $T(n) = O(n)$.
Assume: $T(k) \leq ck, k < n$
Then, $T(n) = T(\sqrt{n}) + n \leq c\sqrt{n} + n = cn - (cn - c\sqrt{n} - n)$. If inequality $cn - c\sqrt{n} - n \geq 0$ always holds true, i.e. $(c-1)n \geq c\sqrt{n}$, we can set $c = 2, n \geq 4$. It means that $T(n) \leq cn$ holds true as long as $c = 2$ and $n \geq 4$. Therefore, our claim is correct!
- *Lower bound.* Claim: $T(n) = \Omega(n)$.
Assume: $T(k) \geq ck, k < n$
Then, $T(n) = T(\sqrt{n}) + n \geq c\sqrt{n} + n = cn - (cn - c\sqrt{n} - n)$. If inequality $cn - c\sqrt{n} - n \leq 0$ holds true, i.e. $(c-1)n \leq c\sqrt{n}$, we can set $c = 0.5, n \geq 1$. It means $T(n) \geq cn$ holds true as long as $c = 0.5$ and $n \geq 1$. Therefore, our claim is correct!
Since $T(n) = O(n), T(n) = \Omega(n)$, it implies that $T(n) = n$.

Problem 4 (10pt)

The recurrence is

$$T(n) = 3T(n/3) + f(n)$$

(1) If want $T(n) = O(n \log n)$.
By looking at master theorem (2nd condition), $n^{\log_b a} = n^{\log_3 3}$, if $f(n) = \Theta(n^{\log_b a}) = \Theta(n)$, then $T(n) = \Theta(n \log n)$. For example, we can take $f(n) = n$.
Caveat: If $f(n) = \Theta(n \log n) \neq \Omega(n^{1+\epsilon})$ for some $\epsilon > 0$, So we cannot apply master theorem (3rd condition) in this case.
(2) If want $T(n) = O(n^2)$, then $f(n) = \Omega(n^{\log_b a + \epsilon})$ for some $\epsilon > 0$, then $T(n) = \Theta(f(n))$. For example, we can take $f(n) = n^2$.