# EL9343 Homework 2

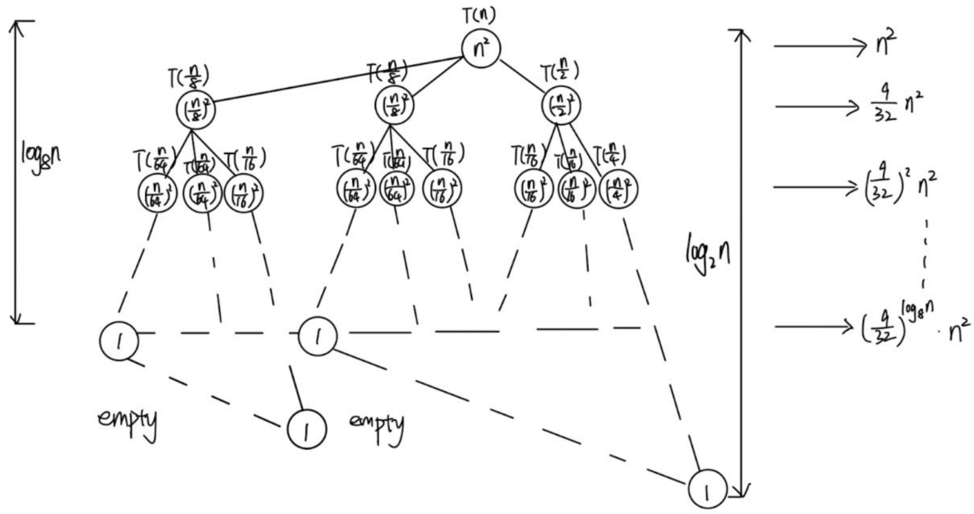
Due: Sept. 22nd 11:00 a.m.

1. First use the iteration method to solve the recurrence, draw the recursion tree to analyze.

$$T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2$$

Then use the substitution method to verify your solution.

Part 1 - the iteration method:



The recursion tree is asymmetric and unbalanced. The minimum depth is $\log_8 n$ at the most left side while the maximum is $\log_2 n$ at the most right side. Note that the cost at each depth is reduced by a factor of $\frac{9}{32}$ before reaching the depth $\log_8 n$. In other words, the merging cost at depth $k$ is bounded by $n^2 \times (\frac{9}{32})^k$. Then we can bound $T(n)$ from above and below by,

$$T_{upper}(n) = (1 + (\frac{9}{32}) + (\frac{9}{32})^2 + \cdots + (\frac{9}{32})^{log_2 n})n^2$$
$$T_{lower}(n) = (1 + (\frac{9}{32}) + (\frac{9}{32})^2 + \cdots + (\frac{9}{32})^{log_8 n})n^2$$
$$T_{lower}(n) \leq T(n) \leq T_{upper}(n)$$

Both $T_{upper}(n)$ and $T_{lower}(n)$ are $\Theta(n^2)$, since $\lim\limits_{n\to\infty} \frac{T_{upper}(n)}{n^2} = \lim\limits_{n\to\infty} \frac{T_{lower}(n)}{n^2} = \frac{32}{23}$.
Therefore, $T(n) = \Theta(n^2)$.

Part 2 - substitution method:
The upper bound:
IH: $T(k) \leq dk^2,\ \forall k < n \implies T(n) \leq dn^2$, so we have:

$$T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2 \leq d(\frac{n}{2})^2 + 2d(\frac{n}{8})^2 + n^2 \leq dn^2 \Leftrightarrow d \geq \frac{32}{23}$$

Thus, if we set $d \geq \frac{32}{23}$, then for all $n$, $T(n) \leq dn^2 \implies T(n) = O(n^2)$.
The lower bound:
IH: $T(k) \geq ck^2,\ \forall k < n \implies T(n) \geq cn^2$, so we have:

$$T(n) = T(\frac{n}{2}) + 2T(\frac{n}{8}) + n^2 \geq c(\frac{n}{2})^2 + 2c(\frac{n}{8})^2 + n^2 \geq cn^2 \Leftrightarrow c \leq \frac{32}{23}$$

Thus, if we set $c \leq \frac{32}{23}$, then for all $n$, $T(n) \geq cn^2 \implies T(n) = \Omega(n^2)$.
With both upper and lower bounds, we can get $T(n) = \Theta(n^2)$.

2. Use the substitution method to prove that,

$$T(n) = 2T(\frac{n}{2}) + cn \log n$$

is $O(n(\log n)^2)$, where $c > 0$ is a constant. ($\log \equiv \log_2$, in this and the following questions)

IH: $T(k) \leq dk(\log k)^2,\ \forall k < n \implies T(n) \leq dn(\log n)^2$, so we have:

$$T(n) = 2T(\frac{n}{2}) + cn \log n \leq 2d\frac{n}{2}(\log \frac{n}{2})^2 + cn \log n \leq dn(\log n)^2$$
$$\Leftrightarrow dn(\log n - 1)^2 + cn \log n \leq dn(\log n)^2$$
$$\Leftrightarrow d \geq \frac{c}{2 - \frac{1}{\log n}}$$

As $n \to \infty$, $\frac{c}{2-\frac{1}{\log n}}$ monotone decreasing. So for sufficiently large $n$, we choose $d \geq c$, then we have:

$$d \geq c \geq \frac{c}{2 - \frac{1}{\log 2}} \geq \frac{c}{2 - \frac{1}{\log n}}$$

Therefore, $T(n) = O(n(\log n)^2)$.

3. Solve the recurrence:

$$T(n) = 2T(\sqrt{n}) + (\log \log n)^2$$

(Hint: How to make change of variables so that you can apply Master's method?)

Let $m = \log n$, so $n = 2^m$, and,

$$T(2^m) = 2T(\sqrt{2^m}) + (\log m)^2$$
$$T(2^m) = 2T((2^m)^{\frac{1}{2}}) + (\log m)^2$$
$$T(2^m) = 2T(2^{\frac{m}{2}}) + (\log m)^2$$

Let $S(m) = T(2^m)$, then $S(m) = 2S(\frac{m}{2}) + (\log m)^2$
By master's method, we have $a = b = 2, d = \log_b a = 1, f(m) = (\log m)^2 = O(m^{d-\epsilon})$, for some $\epsilon > 0$, so $S(m) = \Theta(m)$ and $T(n) = \Theta(\log n)$

4. You want to solve the following three recurrence formulas:

$$\text{A: } T(n) = 5T(\frac{n}{2}) + an$$
$$\text{B: } T(n) = T(\frac{n}{3}) + bn^2$$
$$\text{C: } T(n) = 3T(\frac{n}{3}) + cn \log n$$

Can you use Master's method for each of these? If yes, write down how you check the conditions and the answer. If not, briefly explain why and solve using other methods.
(Hint: You may need the harmonic series, i.e. when $n$ is very large, $\log n \approx \sum_{k=1}^{n} \frac{1}{k}$)
For A, we can use Master's method.

$$a = 5, b = 2, f(n) = \Theta(n)$$

$$\therefore d = \log_b a = \log 5(\approx 2.322), f(n) = O(n^2) = O(n^{d-\epsilon}),\ \text{for } \epsilon = \log 5 - 2 > 0,\ \text{then } T(n) = \Theta(n^d) = \Theta(n^{log5})$$

For B, we can also use Master's method.

$$a = 1, b = 3, f(n) = \Theta(n^2)$$
$$\therefore d = \log_b a = \log_3 1 = 0, f(n) = \Omega(n) = \Omega(n^{d+\epsilon}),\ \text{for } \epsilon = 1 > 0$$
$$\text{Also, } f(n) = \Theta(n^2) \implies \exists c_0 < 1, \forall \text{large } n, af(n/b) \leq c_0 f(n),$$
$$\text{Then, } T(n) = \Theta(f(n)) = \Theta(n^2)$$

For C, we cannot use Master's method.

$$a = 3, b = 3, f(n) = \Theta(n \log n)$$
$$\therefore d = \log_b a = \log_3 3 = 1$$
$$\text{Checking the condition, } f(n) = \Theta(n \log n) \neq \Omega(n^{d+\epsilon}), \text{ for some constant } \epsilon > 0$$

It is also obvious that this recurrence is not for other two cases. So we cannot apply Master's method here.

$$\begin{aligned}
T(n) &= 3T(\frac{n}{3}) + cn \log n \\
&= 9T(\frac{n}{9}) + 3c\frac{n}{3} \log \frac{n}{3} + cn \log n \\
&= 9T(\frac{n}{9}) + cn(\log \frac{n}{3} + \log n) \\
&\dots \\
&= 3^k T(\frac{n}{3^k}) + cn \sum_{i=0}^{k-1} (\log \frac{n}{3^i})
\end{aligned}$$

Assume $T(1) = 1$, and $k \to \log_3 n$, we have,

$$\begin{aligned}
3^k T(\frac{n}{3^k}) &\approx n \\
T(n) &\approx n + cn \sum_{i=0}^{k-1} (\log \frac{n}{3^i}) \\
&= n + cn \sum_{i=0}^{k-1} (\log n - i \log 3) \\
&= n + cnk \log n - c \log 3 \times n \sum_{i=0}^{k-1} i
\end{aligned}$$

Since $k \to \log_3 n$ and $\sum_{i=0}^{k-1} i \approx \frac{k^2}{2} \approx \frac{1}{2}(\log n)^2$, we can see that $T(n) = \Theta(n(\log n)^2)$

**Note1:** Using other methods like substitution method is fine. Big-Theta notation is not required but recommended.

**Note2:** Another way is to use some extension of the Master theorem: if $f(n) = \Theta(n^{\log_b a}(\log n)^k)$, for some constant $k \geq 0$, then $T(n) = \Theta(n^{\log_b a}(\log n)^{k+1})$. You can find this extension in the latest edition of the CLRS book and on some website. There is also further extension about non-positive values of $k$.

**Note3:** For this part C, harmonic series is not needed. It is used when solving $T(n) = 3T(\frac{n}{3}) + \frac{cn}{\log n}$.