Aditya Shirode, Rhythm Shah, Shrenuj Gandhi

4. [8 points, not evenly distributed] *Purpose: application of the Master Theorem and ability to recognize and deal with situations where it does not apply*

   Give Θ bounds for $T(n)$ in each of the following recurrences. Assume $T(n)$ is constant for small values of n. In situations where the solution is based on the Master Theorem,[1] please state which case of the Master Theorem applies. If the Master Theorem does not apply, solve the recurrence using the tree/levels method. You can choose the value of n at which $T(n)$ is a constant, i.e., you can choose the base case for the recursion to be what ever is most convenient. Also, you can let $n = b^k$ for some k.

(a) $T(n) = 15T(n/4) + n^2$

(b) $T(n) = 2T(n/2) + n \lg^2 n$

(c) $T(n) = 4T(n/2) + n^2 \lg \lg n$

(d) $T(n) = 5T(n/4) + \dfrac{n}{\lg^2 n}$

Answer:

(a) Here f(n) is n^2, $a = 15$ and $b = 4$

This makes $(n^{log_b a}) = n^{log_4 15}$

$< n^{log_4 16}$

$= n^2$

$=$ f(n)

This implies f(n) $\in \Omega$ $(n^{log_b a + \epsilon})$, where $\epsilon = 0.5$ and $c = 31/32$

For the regularity condition: $a(f(n/b)) \leq cf(n) \quad \forall\ n \geq n_0$

Hence according to $Case$ 3 of Master's Theorem

T(n) $\in \Theta$ (f(n))

T(n) $\in \Theta$ (n^2)

(b) Here f(n) is $n\ lg^2 n$, $a = 2$ and $b = 2$

This makes $(n^{log_b a}) = n^{log_2 2}$

$= n$

$\neq nlg^2 n$

$=$ f(n) Hence we can rule out $Case$ 2

For $Case$ 1, we need $\epsilon > 0, \ni$ f(n) grows slower than $O(n^{1-\epsilon})$. But for any value of $\epsilon > 0$, f(n) will grow faster than $O(n^{1-\epsilon})$. Hence $Case$ 1 cannot be applied.

Similarly for $Case$ 3, we need $\epsilon > 0, \ni$ f(n) grows faster than $O(n^{1+\epsilon})$. But for any value of $\epsilon > 0$, f(n) will grow slower than $O(n^{1+\epsilon})$. Hence $Case$ 3 cannot be applied.

[1] There is a more general version of the Master Theorem available on the internet. The term "Master Theorem" here refers to the one in the textbook. If you use the more general version, you have to prove it.

Thus *Master Theorem* is not applicable in this case. Hence we use the tree/levels method.

Hence, for $T(n) = 2T(n/2) + n \lg^2 n$
Assume $n = 2^k$, which implies $k = lgn$ and $T(1) = c$

level	instances	instance size	cost/instance	total cost
0	1	n	nlg^2n	nlg^2n
1	2	$n/2$	$\frac{n}{2}lg^2(n/2)$	$nlg^2(n/2)$
2	4	$n/4$	$\frac{n}{4}lg^2(n/4)$	$nlg^2(n/4)$
.	.	.	.	.
i	2^i	$n/2^i$	$\frac{n}{2^i}lg^2(n/2^i)$	$nlg^2(n/2^i)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	2^k	1	c	cn

From the above table we can write

$$\begin{aligned}
\mathrm{T(n)} &= cn + \textstyle\sum_{i=0}^{k-1} nlg^2(n/2^i) \\
&= cn + n\textstyle\sum_{i=0}^{k-1} (lg(n/2^i))^2 \\
&= cn + n\textstyle\sum_{i=0}^{k-1} (lg(n) - lg(2^i))^2 && \text{as } lg(a/b) = lg(a) - lg(b) \\
&= cn + n\textstyle\sum_{i=0}^{k-1} (lg(n))^2 + n\sum_{i=0}^{k-1} (lg(2^i))^2 - n\sum_{i=0}^{k-1} 2(lg(n))(lg(2^i)) \\
& && \text{as } (a-b)^2 = a^2 + b^2 - 2ab \\
&= cn + nlg^2n\textstyle\sum_{i=0}^{k-1} 1 + n\sum_{i=0}^{k-1} i^2 - 2nlgn\sum_{i=0}^{k-1} i && \text{as } lg(2^i) = i \\
&= cn + nlg^2n(k-1) + n\tfrac{(k-1)(k)(2k-1))}{6} - 2nlgn\tfrac{(k-1)(k)}{2} \\
& && \text{as } \textstyle\sum i = \frac{n(n+1)}{2} \textit{ and } \sum i^2 = \frac{n(n+1)(2n+1)}{6} \\
&= cn + nlg^2n(lgn-1) + n\tfrac{(lgn-1)(lgn)(2lgn-1))}{6} - nlgn(lgn-1)(lgn) && \text{as } k = lgn \\
&= \tfrac{1}{3}nlg^3n - \tfrac{1}{2}nlg^2n + \tfrac{1}{6}nlgn + cn
\end{aligned}$$

This means that T(n) $\in \Theta(nlg^3n)$.

(c) Here f(n) is n^2lglgn, $a = 4$ and $b = 2$
This makes $(n^{log_b a}) = n^{log_2 4}$
$= n^2$
$\neq n^2lglgn$
$=$ f(n) Hence we can rule out Case 2

For *Case* 1, we need $\epsilon > 0, \ni$ f(n) grows slower than $O(n^{2-\epsilon})$. But for any value of $\epsilon > 0$, f(n) will grow faster than $O(n^{2-\epsilon})$. Hence *Case* 1 cannot be applied.

Similarly for *Case* 3, we need $\epsilon > 0 \ni$ f(n) grows faster than $O(n^{2+\epsilon})$. But for any value of $\epsilon > 0$ f(n) will grow slower than $O(n^{2+\epsilon})$. Hence *Case* 3 cannot be applied.

Thus $Master'sTheorem$ is not applicable in this case. Hence we use the tree/levels method.

Assume $n = 2^k$, which implies $k = lgn$ and $T(1) = c$

level	instances	instance size	cost/instance	total cost
0	1	n	n^2lglgn	n^2lglgn
1	4	$n/2$	$n^2/4lglg(n/2)$	$n^2lglg(n/2)$
2	4^2	$n/4$	$n^2/4^2lglg(n/4)$	$n^2lglg(n/4)$
.	.	.	.	.
i	4^i	$n/2^i$	$n^2/4^ilglg(n/2^i)$	$n^2lglg(n/2^i)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	$\vdots$
k	4^k	1	c	cn^2

From the above table we can write

T(n) $= cn^2 + \sum_{i=0}^{k-1} n^2lglg(n/2^i)$

$= cn^2 + n^2\sum_{i=0}^{k-1} lglg(\frac{2^k}{2^i})$ as $n = 2^k$

$= cn^2 + n^2\sum_{i=0}^{k-1} lglg(2^{k-i})$

$= cn^2 + n^2\sum_{i=0}^{k-1} lg(k-i)$ as $lg2^i = i$

$= cn^2 + n^2[lg(k) + lg(k-1) + lg(k-2) + .. + lg(1)]$

$= cn^2 + n^2[lg(k.(k-1).(k-2)..1)]$

$= cn^2 + n^2(lg(k!))$

$= cn^2 + n^2klgk$ using Stirling's Approximation $lg(n!) = nlgn - n + O(ln(n))$ and taking its fastest growing term

$= n^2lgnlglgn + cn^2$ as $k = lgn$

This means that T(n) $\in \Theta(n^2lgnlglgn)$.

(d) Here f(n) is $\frac{n}{lg^2n}$, $a = 5$ and $b = 4$

This makes $(n^{log_ba}) = n^{log_45}$
$> n^{log_44}$
$= n$
$> \frac{n}{lg^2n}$
= f(n)

This implies f(n) $\in$ O$(n^{log_ba-\epsilon})$, where $\epsilon = 0.15$

Hence according to *Case* 1 of Master's Theorem,
T(n) $\in \Theta(n^{log_ba})$
T(n) $\in \Theta(n^{log_45})$