CS / MCS 401 Homework 2 grader solutions

assignment due June 27, 2016
written by Jānis Lazovskis
maximum points: 30

Questions from CLRS. Questions marked with an asterisk ∗ were not graded.

3.1-2 *(p.52)* *Show that for any real constants a and b, where $b > 0$, $(n+a)^b = \Theta(n^b)$.*

We claim that there exists n_0 such that for all $n \geqslant n_0$,

$$\left(\frac{1}{2}\right)^b n^b \leqslant (n+a)^b \leqslant \left(\frac{3}{2}\right)^b n^b. \tag{1}$$

This comes from noting that for all $n > 2a$,

$$\frac{1}{2}n \leqslant n + a \leqslant \frac{3}{2}n. \tag{2}$$

Indeed, if $n = 2a + \epsilon$ for some $\epsilon > 0$, then the above inequality becomes

$$a + \frac{1}{2}\epsilon \leqslant 3a + \epsilon \leqslant 3a + \frac{3}{2}\epsilon,$$

which is immediately true if $a \geqslant 0$, and is true for $\epsilon > |3a|$ if $a < 0$. Hence raising all sides of the inequality (2) to a positive power b (which preserves the directions of the inequality for large enough n, as everything is then positive), we get inequality (1), as desired. This is also bounded below by 0 for large enough n, therefore $(n+a)^b$ is $\Theta(n^b)$. ■

3.1-4 *(p.53)* *Is $2^{n+1} = O(2^n)$? Is $2^{2n} = O(2^n)$?*

The first statement is true. Note that $2^{n+1} = 2 \cdot 2^n$, so using a constant of $c = 2$ we see that $2^{n+1} \leqslant 2 \cdot 2^n$ for all n.

The second statement is false. Suppose that $2^{2n} = O(2^n)$, with constant c. Then for all $n \geqslant n_0$ for some fixed n_0, we have that

$$2^{2n} \leqslant c \cdot 2^n = 2^{\log_2(c)} \cdot 2^n = 2^{\log_2(c)+n}.$$

However, since $\log_2(c)$ is finite, there is some $N > \log_2(c)$, for which it is immediate that $2^{2N} > 2^{\log_2(c)+N}$, contradicting the above. Therefore the statement is false. ■

3.2-2 *(p.60)* *Prove equation (3.16), which states that $a^{\log_b(c)} = c^{\log_b(a)}$.*

Use equation (3.15), which states that $\log_x(y) = \ln(x)/\ln(y)$. Applying reversible operations to the equation above, we reach a true statement, and so the original statement is true. Indeed,

$$\begin{aligned}
a^{\log_b(c)} &= c^{\log_b(a)} && \text{(given)}\\
\ln(a^{\log_b(c)}) &= \ln(c^{\log_b(a)}) && \text{(applying ln to both sides)}\\
\log_b(c)\ln(a) &= \log_b(a)\ln(c) && \text{(laws of logarithms)}\\
\frac{\ln(c)\ln(a)}{\ln(b)} &= \frac{\ln(a)\ln(c)}{\ln(b)}, && \text{(equation 3.15)}
\end{aligned}$$

which is true. ■

3.2-8 *(p.60)* *Show that $k\ln(k) = \Theta(n)$ implies $k = \Theta(n/\ln(n))$.*

We are given that $k\ln(k) = \Theta(n)$, or that there exist positive constants c_1, c_2 such that

$$0 \leqslant c_1 n \leqslant k\ln(k) \leqslant c_2 n. \tag{3}$$

Dividing this inequality by $\ln(n)$, we get

$$0 \leqslant c_1 \frac{n}{\ln(n)} \leqslant k\frac{\ln(k)}{\ln(n)} \leqslant c_2 \frac{n}{\ln(n)}. \tag{4}$$

Since $\ln(k)/\ln(n) \to 0$ as $n \to \infty$, we will keep c_1 as the lower bound constant to show that $k = \Theta(n/\ln(n))$ (that is, replacing $\ln(k)/\ln(n)$ with the constant 1 does not change the truth of the middle inequality of (4)).

For the upper bound, use the left side of (3) to note that

$$c_1 n \leqslant k\ln(k) < k^2 \quad\Longrightarrow\quad \ln(c_1) + \ln(n) < 2\ln(k) \quad\Longrightarrow\quad \frac{\ln(n)}{\ln(k)} < 2 - \frac{\ln(c_1)}{\ln(k)} < 2$$

for k large enough. Finally, rewrite k and use the right side of (4) to get

$$k = k\frac{\ln(k)}{\ln(n)}\frac{\ln(n)}{\ln(k)} < c_2\frac{n}{\ln(n)} \cdot 2.$$

Now we have that

$$0 \leqslant c_1 \frac{n}{\ln(n)} \leqslant k \leqslant (2c_2)\frac{n}{\ln(n)}$$

for n large enough, or in other words, that k is $\Theta(n/\ln(n))$. ■

3.2 *(p.61)* *Indicate, for each pair of expressions (A, B) in the table below, whether A is O, o, Ω, ω, or Θ of B. Assume that $k \geqslant 1$, $\epsilon > 0$, and $c > 1$ are constants. Your answer should be in the form of the table with "yes" or "no" written in each box.*

The table is given below. Some justification is expected.

A	B	O	o	Ω	ω	Θ
$\log_2^k(n)$	n^ϵ	yes	yes	no	no	no
n^k	c^n	yes	yes	no	no	no
$\sqrt{n}$	$n^{\sin(n)}$	no	no	no	no	no
2^n	$2^{n/2}$	no	no	yes	yes	no
$n^{\log_2(c)}$	$c^{\log_2(n)}$	yes	no	yes	no	yes
$\log_2(n!)$	$\log_2(n^n)$	yes	no	yes	no	yes

Here are some short justification arguments:

a. Any polynomial grows strictly faster than any logarithm.
b. Any exponential grows strictly faster than any polynomial.
c. The maxima of $n^{\sin(n)}$ increase exponentially, yet it has value 1 periodically. However, the function $\sqrt{n}$ is monotonically increasing past 1, so there are no relations among them in terms of growth.
d. Decreasing the base (from 2 to $\sqrt{2} = 2^{1/2}$) of an exponential function makes it grow strictly slower.
e. Question 3.2-2 above shows both functions are the same.
f. Showing O and Ω involves ounding the sum. For Θ, use Stirling's apporximation on page 57. ■

4.3-1 *(p.87)* *Show that the solution of $T(n) = T(n-1) + n$ is $O(n^2)$.*

We find the answer by expressing $T(n)$ in terms of elements in the seqence that come before it. We find that

$$\begin{aligned} T(n) &= T(n-1) + n \\ &= T(n-2) + (n-1) + n \\ &= T(n-3) + (n-2) + (n-1) + n \\ &\vdots \\ &= T(0) + \sum_{k=1}^{n} k \\ &= T(0) + \frac{n(n+1)}{2} \\ &= O(n^2), \end{aligned}$$

as desired. ■

4.3-2 *(p.87)* *Show that the solution of $T(n) = T(\lceil n/2 \rceil) + 1$ is $O(\log_2(n))$.*

Similarly to above, we simplfy $T(n)$ to find

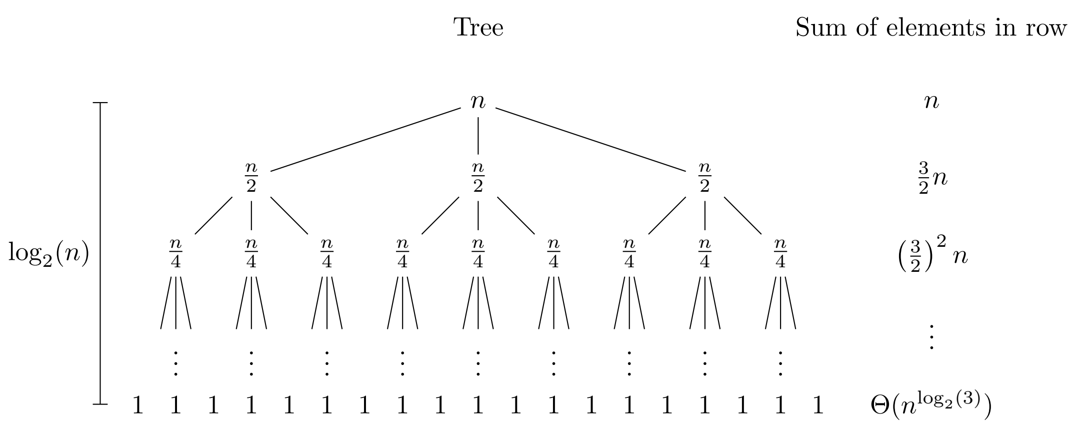
$$T(n) = T(\lceil n/2 \rceil) + 1 = T(\lceil n/4 \rceil) + 2 = T(\lceil n/8 \rceil) + 3 = \cdots = T(1) + c.$$

This terminates at 1 after a certain number of steps, say k. That is, after $n/2^k \leqslant 1$ (because then $1 = \lceil n/2^k \rceil = \lceil n/2^{k+1} \rceil$), or when $n \leqslant 2^k$, or when $k \leqslant \log_2(n)$. Since c is the nmber of steps we've taken, it follows that $c = \lceil \log_2(n) \rceil$. Hence the solution of $T(n)$ is $O(\log_2(n))$. ■

∗ **4.3-6** *(p.87)* *Show that the solution to $T(n) = 2T(\lfloor n/2 \rfloor + 17) + n$ is $O(n \log_2(n))$.*

4.4-1 *(p.92)* *Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = 3T(\lfloor n/2 \rfloor) + n$. Use the substitution method to verify your answer.*

We construct the tree as described in the book, summing all the elements of each row. We make the assumption that n is a power of 2 and we ignore the floor function.



The last row has $(3/2)^{\log_2(n)} n$ elements, but

$$\left(\tfrac{3}{2}\right)^{\log_2(n)} n = n^{\log_2(3/2)} n = n^{\log_2(3) - \log_2(2) + 1} = n^{\log_2(3)},$$

so we get the given result.

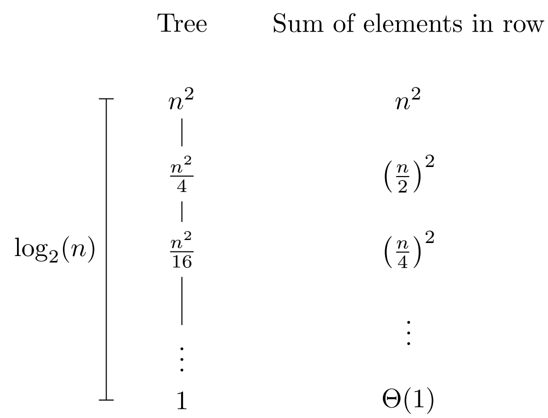
Taking the sum of the elements of each row, we get

$$
\begin{aligned}
T(n) &= n + \frac{3}{2}n + \left(\frac{3}{2}\right)^2 n + \cdots + \left(\frac{3}{2}\right)^{\log_2(n)-1} n + \Theta(n^{\log_2(3)}) \\
&= \sum_{i=0}^{\log_2(n)-1} \left(\frac{3}{2}\right)^i n + \Theta(n^{\log_2(3)}) \\
&= n \cdot \frac{(3/2)^{\log_2(n)} - 1}{3/2 - 1} + \Theta(n^{\log_2(3)}) \\
&= 2n\left(\frac{3^{\log_2(n)}}{2^{\log_2(n)}} - 1\right) + \Theta(n^{\log_2(3)}) \\
&= \frac{2n \cdot n^{\log_2(3)}}{n} - 2n + \Theta(n^{\log_2(3)}) \\
&= n^{\log_2(3)} - 2n + \Theta(n^{\log_2(3)}) \\
&= O(n^{\log_2(3)}),
\end{aligned}
$$

where in the third line we used the formula for a finite geometric series (equation (A.5) on page 1147). This is our asymptotic upper bound on the recurrence $T(n)$. ■

4.4-2 *(p.92)* *Use a recursion tree to determine a good asymptotic upper bound on the recurrence $T(n) = T(n/2)+n^2$. Use the substitution method to verify your answer.*

This is the tree corresponding to the recurrence:



The last line, by the pattern of the sum of elements in each row, should have $\frac{n^2}{2^{2\log_2(n)}} = \frac{n^2}{n^2} = 1$ element. Taking the sum of the elements of each row, we get

$$
\begin{aligned}
T(n) &= n^2 + \left(\frac{n}{2}\right)^2 + \left(\frac{n}{4}\right)^2 + \cdots + \left(\frac{n}{2^{\log_2(n)-1}}\right)^2 + \Theta(1) \\
&= \sum_{i=0}^{\log_2(n)-1} \frac{n^2}{2^{2i}} + \Theta(1) \\
&= n^2 \cdot \frac{(1/4)^{\log_2(n)} - 1}{1/4 - 1} + \Theta(1) \\
&= -3n^2\left(\frac{1}{n^2} - 1\right) + \Theta(1) \\
&= 3n^2 - 3 + \Theta(1) \\
&= O(n^2).
\end{aligned}
$$

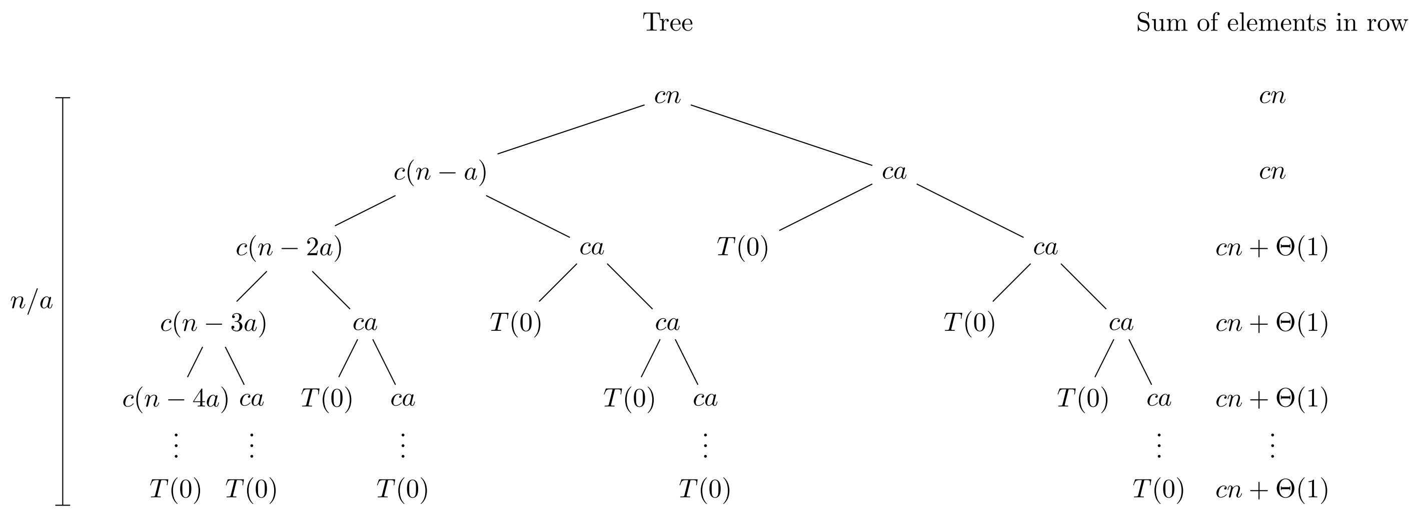
Note the constant -3 is absorbed into $\Theta(1)$, giving us an asymptotic upper bound of $O(n^2)$. Note that this is actually an asymptotically tight bound, so we could have written $\Theta(n^2)$ as well. ■

$*$ **4.4-4** *(p.93)* *Use a recursion tree to determine a good asymptotic upper bound on the recurrence* $T(n) = 2T(n-1)+1$.

$*$ **4.4-5** *(p.93)* *Use a recursion tree to determine a good asymptotic upper bound on the recurrence* $T(n) = T(n-1)+T(n/2)+n$.

4.4-8 *(p.93)* *Use a recursion tree to give an asymptotically tight solution to the recurrence* $T(n) = T(n-a) + T(a) + cn$, *where* $a \geqslant 1$ *and* $c > 0$ *are constants.*

This is the tree corresponding to the recurrence:



Taking the sum of the elements of each row, we get

$$\begin{aligned} T(n) &= cn + cn + cn + \Theta(1) + \cdots + cn + \Theta(1) \\ &= (n/a) \cdot cn + \Theta(1) \\ &= (c/a)n^2 + \Theta(1) \\ &= \Theta(n^2), \end{aligned}$$

which is asymptotically tight because we did not introduce any slopiness into the calculations. ■

$*$ **4.4-9** *(p.93)* *Use a recursion tree to give an asymptotically tight solution to the recurrence* $T(n) = T(\alpha n) + T((1-\alpha)n) + cn$, *where* α *is a constant in the range* $0 < \alpha < 1$ *and* $c > 0$ *is also a constant.*

4.5-1 *(p.96)* *Use the master method to give tight asymptotic bounds for the following recurrences.*

For all these $a = 2$ and $b = 4$, and $\log_4(2) = 1/2$.

a. $T(n) = 2T(n/4) + 1$.
Since 1 is a constant, we can only say it is $O(n^{1/2-\epsilon})$, so case 1 applies, and $T(n) = \Theta(n^{1/2})$.

b. $T(n) = 2T(n/4) + \sqrt{n}$.
Since $\sqrt{n} = \Theta(n^{1/2})$, case 2 applies, and $T(n) = \Theta(n^{1/2}\log_2(n))$.

c. $T(n) = 2T(n/4) + n$.
Since $n = \Omega(n^{1/2+\epsilon})$ and $2 \cdot n/4 = 2n \leqslant 3n$, where $3 > 1$ is certainly a constant, case 3 applies, and $T(n) = \Theta(n)$.

d. $T(n) = 2T(n/4) + n^2$.
Since $n = \Omega(n^{1/2+\epsilon})$ and $2 \cdot (n/4)^2 = n^2/8 \leqslant 2n^2$, where $2 > 1$ is certainly a constant, case 3 applies, and $T(n) = \Theta(n^2)$. ■

4.5-3 *(p.97)* *Use the master method to show that the solution to the binary-search recurrence* $T(n) = T(n/2) + \Theta(1)$ *is* $T(n) = \Theta(\log_2(n))$.

Here $a = 1$ and $b = 2$, so $\log_2(1) = 0$. Case 2 applies, because $\Theta(n^{\log_2(1)}) = \Theta(n^0) = \Theta(1)$, which is exactly the $f(n)$ term. Hence by the master method, $T(n) = \Theta(n^{\log_2(1)} \log_2(n)) = \Theta(\log_2(n))$, as desired. ■

4.5-4 *(p.97)* *Can the master method be applied to the recurrence* $T(n) = 4T(n/2) + n^2 \log_2(n)$*? Why or why not? Give an asymptotic upper bound for this reccurence.*

Here $a = 4$ and $b = 2$, so $\log_2(4) = 2$. Case 1 does not apply, because $n^2 \log_2(n)$ is not bounded above by $n^{2-\epsilon}$ (that is, it is not $O(n^{2-\epsilon})$. Case 2 does not apply, because it is not bound above or below by n^2 (that is, it is not $\Theta(n^2)$). Finally, case 3 also does not apply, because although $n^2 \log_2(n) = \Omega(n^2)$, it is not $\Omega(n^{2+\epsilon})$, because any positive power of n eventually grows faster than $\log_2(n)$. Hence the master method can not be applied.

An asymptotic upper bound of $O(n^2 \log_2(n))$ may be found via the substitution or recurrence tree method. ■