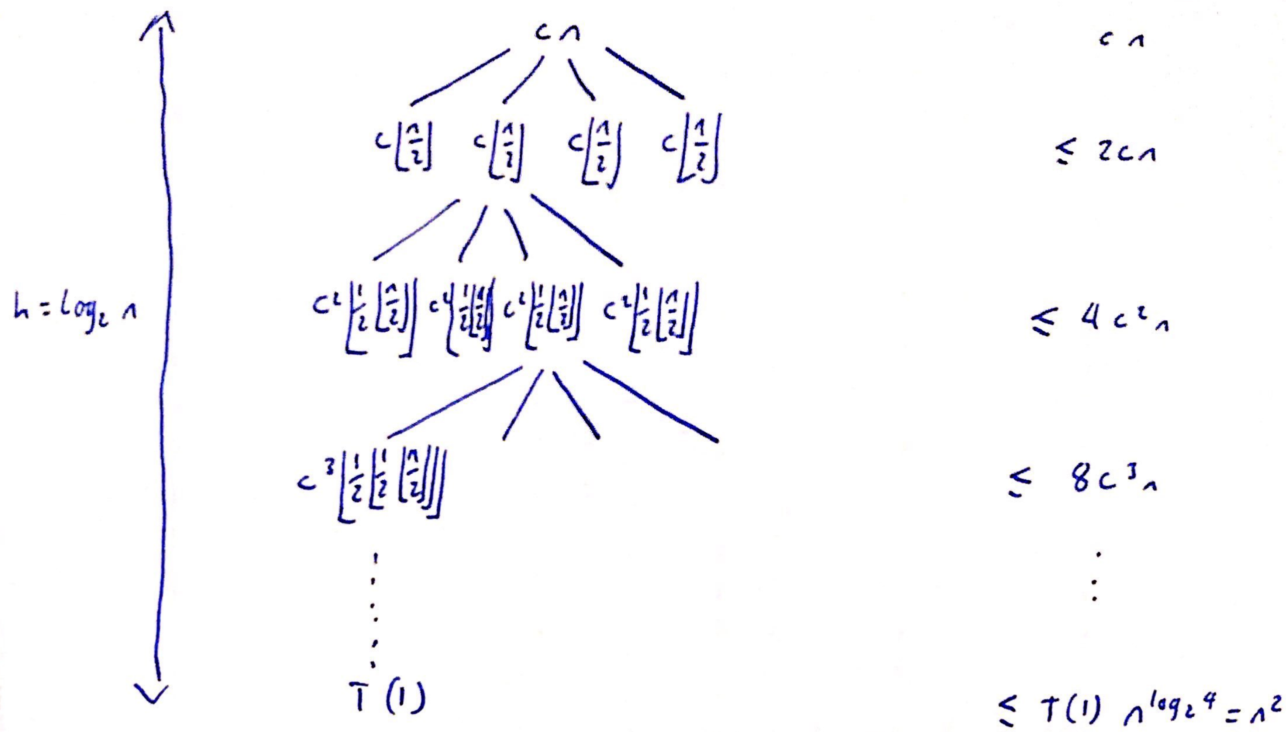
Homework 11

Samuel Pegg

November 2020

1 Page 93 4.4-7

Draw the recursion tree for $T(n) = 4T\left(\left\lfloor \frac{n}{2} \right\rfloor\right) + cn$, where c is a constant, and provide a tight asymptotic bound on its solution. Verify your bound by the substitution method.



The kth layer has complexity $2^k c^k n$, the height of the tree is $\log_2(n)$ and the final layer has $n^{\log_2 4} = n^2$ leaves. Comparing $f(n) = cn$ and n^2 its clear that n^2 grows more quickly. We can write

$$f(n) = O(n^{2-\epsilon})$$

for $\epsilon = 0.5$. Thus

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

To verify this with the substitution method, we need to show $T(n) = O(n^2) = \Omega(n^2)$. Consider $O(n^2)$ first, that is, assume $T(n) \leq an^2 - bn$. Then

$$T(n) \leq 4a\left\lfloor \frac{n}{2} \right\rfloor^2 - 4b\left\lfloor \frac{n}{2} \right\rfloor + cn \leq an^2 - 2bn + cn = an^2 - (2b - c)n = an^2 - bn$$

if we define $b = c$. For $\Omega(n^2)$, assume $T(n) \geq an^2 - bn$. Then

$$T(n) \geq 4a\left\lfloor \frac{n}{2} \right\rfloor^2 - 4b\left\lfloor \frac{n}{2} \right\rfloor + cn \geq 4a\left(\frac{n^2}{4} - n + 1\right) - 4b\left(\frac{n}{2} - 1\right) + cn \geq an^2 - (4a + 2b - c)n = an^2 - bn$$

if we define $b = \frac{c}{2}$ and $a = \frac{c}{8}$. Thus $T(n) = \Omega(n^2) = O(n^2) \implies T(n) = \Theta(n^2)$.

2 Page 107 4-1

Give asymptotic upper and lower bounds for $T(n)$ in each of the following recurrences. Assume that $T(n)$ is constant for $n \leq 2$. Make your bounds as tight as possible and justify your answers.

2.1 b

$T(n) = T\left(\frac{7n}{10}\right) + n$, hence $f(n) = n$ and $n^{\log_{\frac{10}{7}}(1)} = n^0 = 1$. $f(n)$ grows polynomially faster than 1, that is, we can write

$$f(n) = \Omega(n^{0+\epsilon})$$

for $\epsilon = 0.5$. In addition,

$$f\left(\frac{7n}{10}\right) = \frac{7n}{10} \leq cn = cf(n)$$

for $\frac{7}{10} \leq c < 1$, hence

$$T(n) = \Theta(n)$$

2.2 c

$T(n) = 16T\left(\frac{n}{4}\right) + n^2$, hence $f(n) = n^2$ and $n^{\log 4(16)} = n^2$. So $f(n) = \Theta(n^2)$ and n^2 grow at similar rates. Hence

$$T(n) = \theta(n^2 \lg(n))$$

2.3 d

$T(n) = 7T\left(\frac{n}{3}\right) + n^2$, hence $f(n) = n^2$ and $n^{\log_3(7)} \approx n^{1.77}$. We can write

$$f(n) = \Omega(n^{1.77+\epsilon})$$

for $\epsilon = 0.01$, and in addition

$$7f\left(\frac{n}{3}\right) = \frac{7n^2}{9} \leq cn^2 = cf(n)$$

for $\frac{7}{9} \leq c < 1$. hence

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

3 Page 107 4-2

Throughout this book, we assume that parameter passing during procedure calls takes constant time, even if an N-element array is being passed. This assumption is valid in most systems because a pointer to the array is passed, not the array itself. This problem examines the implications of three parameter-passing strategies:

1. *An array is passed by pointer. Time= $\Theta(1)$.*
2. *An array is passed by copying. Time= $\Theta(N)$, where N is the size of the array.*
3. *An array is passed by copying only the subrange that might be accessed by the called procedure. Time= $\Theta(q - p + 1)$ if the subarray $A[p..q]$ is passed.*

3.1 b

For MergeSort,

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

1. $T(n) = 2T\left(\frac{n}{2}\right) + cn$ is $\Theta(n\log(n))$ by the Master Method.
2. $T(n) = 2T\left(\frac{n}{2}\right) + cn + 2N$, so

$$T(n) = \Theta(n^{\lg(2)}) + \sum_{k=0}^{\lg(n)-1} \left\{cn + 2^k N\right\} = \Theta(n) + cn\lg(n) + N\frac{1 - 2^{\lg(n)}}{1-2}$$

$$= \Theta(n) + \Theta(n\lg(n)) + nN - N = \Theta(nN) = \Theta(n^2)$$

3. $T(n) = 2T\left(\frac{n}{2}\right) + cn + 2\frac{n}{2} = 2T\left(\frac{n}{2}\right) + n(c+1)$ is $\Theta(n\lg(n))$ by the master method.