

CSE 3318 Notes 4: Recurrences

(Last updated 8/1/22 2:59 PM)

CLRS 4.3, 4.4

4.A. BOUNDING RECURRENCES ASYMPTOTICALLY

Goal: Take a function $T(n)$ that is defined recursively and find $f(n)$ such that $T(n) \in \Theta(f(n))$.

Need to establish both $T(n) \in O(f(n))$ and $T(n) \in \Omega(f(n))$ (without using the limit theorems).

4.B. RECURRENCES, CONSTANTS, AND EVALUATING BY BRUTE EXPANSION

Consider the recurrence:

$$T(n) = T\left(\frac{n}{2}\right) + e = T\left(\frac{n}{2}\right) + \Theta(1)$$

$$T(1) = d = \Theta(1) \quad [\text{or use an arbitrary } n_0, \text{ i.e. } T(n_0) = d = \Theta(1)]$$

Suppose $n = 2^k$:

$$T(1) = d$$

$$T(2) = T(1) + e = d + e$$

$$T(4) = T(2) + e = (d + e) + e = d + 2e$$

$$T(8) = T(4) + e = (d + 2e) + e = d + 3e$$

$$T(2^k) = T(2^{k-1}) + e = d + ke = d + e \lg n = \Theta(\lg n)$$

Consider the recurrence:

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

$$T(1) = d = \Theta(1)$$

Suppose $n = 2^k$:

$$T(1) = d$$

$$T(2) = 2T(1) + 2e = 2d + 2e$$

$$T(4) = 2T(2) + 4e = 2(2d + 2e) + 4e = 4d + 8e$$

$$T(8) = 2T(4) + 8e = 2(4d + 8e) + 8e = 8d + 24e$$

$$T(16) = 2T(8) + 16e = 2(8d + 24e) + 16e = 16d + 64e$$

$$T(n = 2^k) = 2T(2^{k-1}) + 2^k e = 2(2^{k-1}d + 2^{k-1}(k-1)e) + 2^k e$$

$$= 2^k d + 2^k k e = nd + en \lg n = \Theta(n \log n)$$

(See <https://ranger.uta.edu/~weems/NOTES3318/notes04.c> for examples of expansion)

Constants d and e rarely matter . . . certainly not for $T(n) = aT\left(\frac{n}{b}\right) + f(n)$ recurrences

4.C. THE SUBSTITUTION METHOD FOR BOUNDING RECURRENCES

Method

Guess bound (lower Ω and/or upper O)

[On exams the guess will be given to you]

Verify by math induction (solve for constants for some function in asymptotic set)

i) Assume bounding hypothesis works for $k < n$ [Never, ever $k \leq n$]

ii) Show bounding hypothesis works for n in exactly the same form as (i).

Refine bound

Example: Binary search recurrence (number of probes)

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$O(\log n)$$

Assume $T(k) \leq c \lg k$ for $k < n$. (Note: $\log_a n \in \Theta(\log_b n)$)

$$T\left(\frac{n}{2}\right) \leq c \lg\left(\frac{n}{2}\right) = c(\lg n - \lg 2) = c \lg n - c$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$\leq c \lg n - c + 1$$

$$\leq c \lg n \text{ if } c \geq 1$$

$$-c + 1 \leq 0$$

$\Omega(\log n)$

Assume $T(k) \geq c \lg k$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \lg\left(\frac{n}{2}\right) = c(\lg n - \lg 2) = c \lg n - c$$

$$T(n) = T\left(\frac{n}{2}\right) + 1$$

$$\geq c \lg n - c + 1$$

$$\geq c \lg n \text{ if } 0 < c \leq 1$$

Example: $T(n) = 2T\left(\frac{n}{2}\right) + dn$

$O(n \log n)$

Assume $T(k) \leq c k \lg k$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n}{2} \lg\left(\frac{n}{2}\right) = c \frac{n}{2} (\lg n - \lg 2) = c \frac{n}{2} \lg n - c \frac{n}{2}$$

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

$$\leq 2\left(c \frac{n}{2} \lg n - c \frac{n}{2}\right) + dn$$

$$= cn \lg n - cn + dn$$

$$\leq cn \lg n \text{ if } c \geq d$$

$\Omega(n \log n)$

Assume $T(k) \geq c k \lg k$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \frac{n}{2} \lg\left(\frac{n}{2}\right) = c \frac{n}{2} (\lg n - \lg 2) = c \frac{n}{2} \lg n - c \frac{n}{2}$$

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

$$\geq 2\left(c \frac{n}{2} \lg n - c \frac{n}{2}\right) + dn$$

$$= cn \lg n - cn + dn$$

$$\geq cn \lg n \text{ if } 0 < c \leq d$$

Why is there no basis step for the math induction in these two examples?

$$T(1) \leq c$$

Example: $T(n) = 4T\left(\frac{n}{2}\right) + n$

Try $O(n^3)$ and confirm by math induction:

Assume $T(k) \leq ck^3$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n^3}{8}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

$$\leq 4c \frac{n^3}{8} + n$$

$$= \frac{c}{2} n^3 + n$$

$$= cn^3 - \frac{c}{2} n^3 + n \quad -\frac{c}{2} n^3 + n \leq 0 \text{ if } n \text{ is sufficiently large}$$

$$\leq cn^3$$

Improve bound to $O(n^2)$ and confirm:

Assume $T(k) \leq ck^2$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n^2}{4}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n \leq 4c \frac{n^2}{4} + n = cn^2 + n \text{ STUCK!}$$

$$\leq cn^2$$

~~$cn^2 + n$~~

$\Omega(n^3)$ as lower bound:

Assume $T(k) \geq ck^3$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \frac{n^3}{8}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n$$

$$\geq 4c \frac{n^3}{8} + n$$

$$= \frac{c}{2} n^3 + n$$

$$= cn^3 - \frac{c}{2} n^3 + n \quad -\frac{c}{2} n^3 + n \geq 0?$$

$$\geq cn^3 \quad \text{DID NOT PROVE!!!}$$

$\Omega(n^2)$ as lower bound:

Assume $T(k) \geq ck^2$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \frac{n^2}{4}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n \geq 4c \frac{n^2}{4} + n = cn^2 + n \geq cn^2 \quad \text{for } 0 < c$$

What's going on? May use either brute expansion (or a recursion tree, see 4.D)

$$T(1) = d$$

$$T(2) = 4T(1) + 2 = 4d + 2$$

$$T(4) = 4T(2) + 4 = 4(4d + 2) + 4 = 16d + 8 + 4 = 16d + 12$$

$$T(8) = 4T(4) + 8 = 4(16d + 12) + 8 = 64d + 48 + 8 = 64d + 56$$

$$T(16) = 4T(8) + 16 = 4(64d + 56) + 16 = 256d + 224 + 16 = 256d + 240$$

$$\text{Hypothesis: } T(n) = dn^2 + n^2 - n = (d+1)n^2 - n = cn^2 - n = \Theta(n^2)$$

$O(n^3)$

$\Omega(n^2)$

$O(n^2)$:

Assume $T(k) \leq ck^2 - k$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n^2}{4} - \frac{n}{2}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n \leq 4\left(c \frac{n^2}{4} - \frac{n}{2}\right) + n = cn^2 - 2n + n = cn^2 - n$$

$\Omega(n^2)$:

[This bound was already proven.]

Assume $T(k) \geq ck^2 - k$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \frac{n^2}{4} - \frac{n}{2}$$

$$T(n) = 4T\left(\frac{n}{2}\right) + n \geq 4\left(c \frac{n^2}{4} - \frac{n}{2}\right) + n = cn^2 - 2n + n = cn^2 - n$$

Example: $T(n) = T(\sqrt{n}) + 1$

$O(\log n)$:

Assume $T(k) \leq c \lg k$ for $k < n$

$$T(\sqrt{n}) \leq c \lg \sqrt{n} = c \frac{\lg n}{2}$$

$$T(n) = T(\sqrt{n}) + 1 \leq c \frac{\lg n}{2} + 1 = c \lg n - c \frac{\lg n}{2} + 1$$

$\leq c \lg n$ for sufficiently large n

$O(\log \log n)$:

Assume $T(k) \leq c \lg \lg k$ for $k < n$. (Note: $\log_a \log_a n \in \Theta(\log_b \log_b n)$)

$$T(\sqrt{n}) \leq c \lg \lg \sqrt{n} = c \lg \frac{\lg n}{2} = c \lg \lg n - c$$

$$T(n) = T(\sqrt{n}) + 1 \leq c \lg \lg n - c + 1$$

$$\leq c \lg \lg n \text{ if } c \geq 1$$

Achieving $\Theta(\log \log n)$ seems irrelevant, but ...

2^n is to n as n is to $\log n$... and as $\log n$ is to $\log \log n$.

$\Omega(\log \log n)$:

Assume $T(k) \geq c \lg \lg k$ for $k < n$

$$T(\sqrt{n}) \geq c \lg \lg \sqrt{n} = c \lg \frac{\lg n}{2} = c \lg \lg n - c$$

$$T(n) = T(\sqrt{n}) + 1 \geq c \lg \lg n - c + 1$$

$$\geq c \lg \lg n \text{ if } 0 < c \leq 1$$

Example: $T(n) = 2T\left(\frac{n}{2}\right) + n^3$

$O(n^3)$:

Assume $T(k) \leq ck^3$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n^3}{8}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n^3 \leq 2c \frac{n^3}{8} + n^3 = c \frac{n^3}{4} + n^3$$

$$= cn^3 - \frac{3}{4}cn^3 + n^3$$

$$\leq cn^3 \text{ if } c \geq \frac{4}{3}$$

$\Omega(n^3)$:

Assume $T(k) \geq ck^3$ for $k < n$

$$T\left(\frac{n}{2}\right) \geq c \frac{n^3}{8}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n^3 \geq 2c \frac{n^3}{8} + n^3 = c \frac{n^3}{4} + n^3$$

$$= cn^3 - \frac{3}{4}cn^3 + n^3$$

$$\geq cn^3 \text{ if } 0 < c \leq \frac{4}{3}$$

Alternative proof of $O(n^3)$:

Assume $T(k) \leq ck^3$ for $k < n$

$$T\left(\frac{n}{2}\right) \leq c \frac{n^3}{8}$$

$$T(n) = 2T\left(\frac{n}{2}\right) + n^3 \leq 2c \frac{n^3}{8} + n^3 = c \frac{n^3}{4} + n^3$$

$$= n^3 \left(\frac{c}{4} + 1 \right)$$

$$\leq cn^3 \text{ if } c \geq \frac{4}{3} \text{ which is derived by: } \frac{c}{4} + 1 \leq c$$

$$1 \leq \frac{3}{4}c$$

$$\frac{4}{3} \leq c$$

$$c \frac{n^3}{4} + n^3$$

$$n^3 \left(\frac{c}{4} + 1 \right)$$

Example: $T(n) = 3T\left(\frac{n}{3}\right) + 2$

$O(n)$:

Assume $T(k) \leq ck$ for $k < n$

$$T\left(\frac{n}{3}\right) \leq \frac{cn}{3}$$

$$T(n) = 3T\left(\frac{n}{3}\right) + 2 \leq 3 \frac{cn}{3} + 2 = cn + 2 \text{ stuck!}$$

$\Omega(n)$:

Assume $T(k) \geq ck$ for $k < n$

$$T\left(\frac{n}{3}\right) \geq \frac{cn}{3}$$

$$T(n) = 3T\left(\frac{n}{3}\right) + 2 \geq 3 \frac{cn}{3} + 2 = cn + 2 \geq cn$$

Examine a few cases:

$$T(1) = d$$

$$T(3) = 3T(1) + 2 = 3d + 2$$

$$T(9) = 3T(3) + 2 = 3(3d + 2) + 2 = 9d + 6 + 2 = 9d + 8$$

$$T(27) = 3T(9) + 2 = 3(9d + 8) + 2 = 27d + 24 + 2 = 27d + 26$$

$$\text{Hypothesis: } T(n) = nd + n - 1 = (d+1)n - 1 = cn - 1 = \Theta(n)$$

New try for $O(n)$:

Assume $T(k) \leq ck - 1$ for $k < n$

$$T\left(\frac{n}{3}\right) \leq \frac{cn}{3} - 1$$

$$T(n) = 3T\left(\frac{n}{3}\right) + 2 \leq 3\left(\frac{cn}{3} - 1\right) + 2 = cn - 3 + 2 = cn - 1$$

4.D. RECURSION TREE METHOD

Concepts

Draw tree – either for a particular n or for general case

Fan-out

Sub-problem sizes

Number of levels in tree

Number of leaves

(Analysis at parents-of-leaves level is OPTIONAL!)

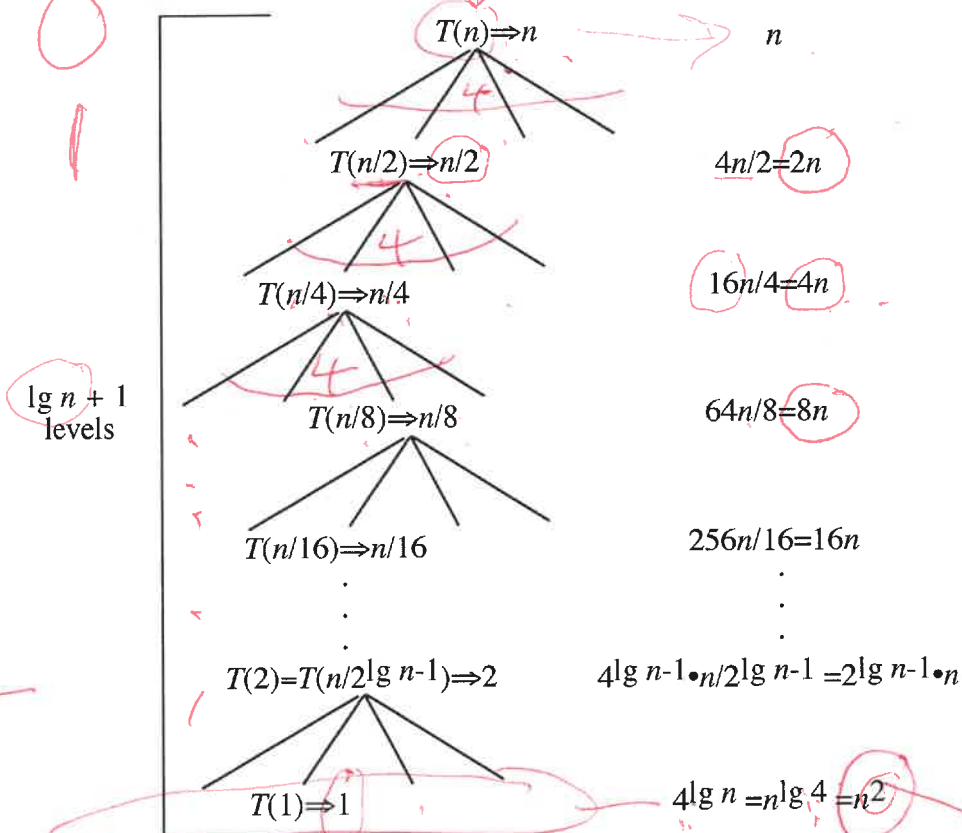
Assign (non-recursive) contribution of a node (sub-problem) in each level

Compute total (non-recursive) contribution across each level [different from brute expansion]

Usually complete by evaluating a summation – Go directly for Θ bound, not separate O and Ω .

Example: $T(n) = 4T\left(\frac{n}{2}\right) + n$

$n = 2^k$



Note use of the identity $b^{\log_a d} = d^{\log_a b}$. (CLRS, p. 66)

Using definite geometric sum formula:

$$n \sum_{k=0}^{\lg n - 1} 2^k + n^2 = n \frac{2^{\lg n} - 1}{2 - 1} + n^2$$

$$= n(n - 1) + n^2$$

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

Using $\sum_{k=0}^t x^k = \frac{x^{t+1} - 1}{x - 1}$ $x \neq 1$

$(\lg n) - 1$
 ~~$\lg(n-1)$~~

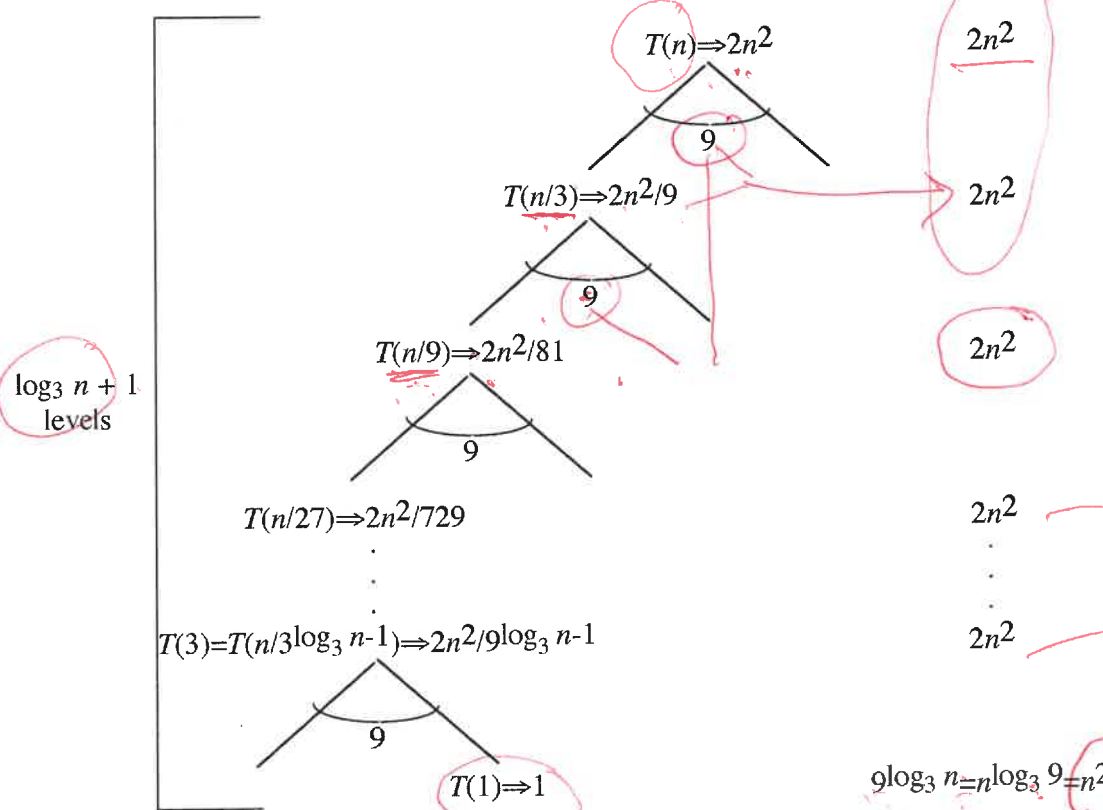
Using definite geometric sum formula:

$$\begin{aligned}
 n^3 \sum_{k=0}^{\lg n - 1} \frac{1}{4^k} + n &= n^3 \frac{\left(\frac{1}{4}\right)^{\lg n} - 1}{\frac{1}{4} - 1} + n \\
 &= n^3 \frac{n^{-2} - 1}{-\frac{3}{4}} + n = n^3 \frac{1 - n^{-2}}{\frac{3}{4}} + n = \frac{4}{3} n^3 \left(1 - \frac{1}{n^2}\right) + n \\
 &= \Theta(n^3)
 \end{aligned}$$

Using $\sum_{k=0}^t x^k = \frac{x^{t+1} - 1}{x - 1}$ $x \neq 1$

Example: $T(n) = 9T\left(\frac{n}{3}\right) + 2n^2$

$n \Rightarrow 3^k$



$$T(n) = 2n^2 \log_3 n + n^2 = \Theta(n^2 \log n)$$

$\log_3 n - 1$
 $\sum_{k=0}^{\log_3 n - 1} 2n^2$

$$9^{\log_3 n} = n^{\log_3 9} = n^2$$