COE428 Notes Week 4 (Week of Jan 30, 2017)

Table of Contents

Announcements

- Midterm: Wednesday, March 8, 2017
- Midterm material **(Subject to change!):**
  - From CLRS textbook:
    - Chapter: 2 (**Getting Started** , 3.1 Insertion sort, 3.2 Analyzing algorithms, 3.3 Designing algorithms)
    - Chapter 3 (**Growth of functions**, 3.1 Asymptotic notation, 3.2 Standard notations and common functions)
    - Chapter 4 (**Divide-and-Conquer,** (4.3 The substitution method for solving recurrences, 4.4 The recursion-tree method for solving recurrences)
  - My lecture notes: Week 1–6
  - Labs: lab 1–5
  - My book:
    - Chapter 1 (**Algorithms**)
    - Chapter 2 (**Recursion**)
    - Chapter 4 (**Complexity**)

Answers to last week's questions

1. Fill in the columns labelled $f(n) = O(g(n))$, $f(n) = \Omega(g(n))$ and $f(n) = \Theta(g(n))$ as true or false.

$f(n)$	$g(n)$	$f(n) = O(g(n))$?	$f(n) = \Omega(g(n))$?	$f(n) = \Theta(g(n))$?
$2^{\lg n} + 5$	n	**true**	**true**	**true**
$1.001^n + 6n^3$	1.2^n	**true**	**false**	**false**
$1.001^n + 6n^3$	n^{1000}	**false**	**true**	**false**
$3 \times 4^{\lg n} + 5n\sqrt{n}$	$n^{1.5}$	**false**	**true**	**false**
$3 \times 4^{\lg n} + 5n\sqrt{n}$	n^2	**true**	**true**	**true**
$\log n!$	$n \log n + n$	**true**	**true**	**true**

2. Determine the Big-Oh complexity of $T(n) = 2T(n/2) + n/\lg n$. (You may find this challenging!) (Note: valid only for $n \geq 2$. You may assume any convenient base case. You may also assume that *n* is a power of 2.)

   Answer: See below where we examine this case in detail in Friday's lecture.

3. Determine the simplest Big-Theta complexity of each the functions below by inspection.

   a) $32 + n\log_5 n + n^2\sqrt{n} = \Theta(n^{3.5})$
   b) $1000000n + 5 \times 2^{n^2} + 123 \times 3^n = \Theta(2^{n^2})$
   c) $5n^3 + \left(\sum_{i=1}^{n} i^2\right)^2 + 20n^5 = \Theta(n^6)$
   d) $\log_{10} n! + n^2 \lg n = \Theta(n^2 \log n)$
   e) $\left(\sum_{i=1}^{n} 1/i\right)^{1.2} + 15\lg n = \Theta(\ln^{1.2} n)$

Review

Growth of functions

- It is easy to find the fastest growing term for polynomials.
- The following list shows examples from slowest growing to fastest growing functions:
  - $5 = \Theta(1)$
  - $23 + 7\log\log n = \Theta(\log\log n)$
  - $7\ln n + 8\lg n + 3\log\log n + 8 = \Theta(\log n)$
  - $\sum_{i=1}^{n} 1/i + \log\log n = \Theta(\log n)$
  - $\log n + \log\log n + \log^2 n = \Theta(\log^2 n)$
  - $n\log n! + n\log n = \Theta(n^2\log n)$
  - $n^{100} + 1.01^n = \Theta(1.01^n)$

Big-O, Big-Omega and Big-Theta analysis of recurrences

Prove that $T(n) = 2T(n/2) + n = O(n^2)$

- (Yes, this seems silly. We have already rigorously proved that $T(n) = 2T(n/2) + n = n \lg n$ so we already **know** that $T(n) = O(n^2)$; indeed we know that $T(n) = O(n \log n)$ and $T(n) = \Theta(n \log n)$) But this simple example illustrates the technique for using mathematical induction to prove an *inequality* instead of an equality as done previously.)
- We start with:
  - A base case: $T(1) = \Theta(1)$
  - Our hypothesis: $T(n) \leq cn^2 \qquad \forall n > n_0$ for some c, n_0
- Our Inductive Hypothesis is $T(m) \leq cm^2$ for all $m < n$.
  - So: $T(n) = 2T(n/2) + n$ (by definition)
  - Using the hypothesis, we get: $T(n) \leq 2(c(n/2)^2 + n$
  - Simplifying we get: $T(n) \leq cn^2/2 + n$
  - We wish to prove our hypothesis: $T(n) \leq cn^2$
  - So we need to prove: $cn^2/2 + n \leq cn^2$ for all $n > n_0$ for some constants n_0, c
  - Let's re-write this as: $cn^2/2 + cn^2/2 - cn^2/2 + n = cn^2 - n \leq cn^2$
  - Or, show that $cn^2 - n \leq cn^2$
  - Or, show that we have to prove: *hypothesis – extra* $\leq$ *hypothesis*
  - Or, simply prove that *extra* $\geq$ *0.*
  - In this case, find some n_0 such that for all $n > n_0$.
  - This is clearly true if $n_0 = 0$

Prove that $T(n) = 2T(n/2) + n = O(n \log n)$

- We start with:
  - A base case: $T(1) = \Theta(1)$
  - Our hypothesis: $T(n) \leq c'n \log n \qquad \forall n > n_0$ for some c, n_0
- Our Inductive Hypothesis is $T(m) \leq c'm \log m = cm \lg m$ for all $m < n$.
  - So: $T(n) = 2T(n/2) + n$ (by definition)
- Using the inductive hypothesis, we get:

$$\begin{aligned}T(n) &\leq 2(cn/2 \lg n/2) + n\\ &= cn \lg(n/2) + n\\ &= cn(\lg n - \lg 2) + n\\ &= cn(\lg n - 1) + n\\ &= cn \lg n - cn + n\\ &= cn \lg n - (cn - n)\end{aligned}$$

- Recall we wish to prove $T(n) \leq cn \lg n$ so we must prove:
  - $T(n) = cn \lg n - (cn - n) \leq cn \lg n$
  - OR *hypothesis – extra <= hypothesis*
  - In short, prove "*extra >= 0*"
  - Here "extra" is $cn - n = n(c-1)$
  - So we need to show that $n(c-1) \geq 0$.
  - Any $c > 1$ and any $n > 0$ will do.
  - **QED!**

Prove that *T(n) = 2T(n/2) + n = Ω(n* log *n)*

- We start with:
  - A base case: $T(1) = \Theta(1)$
  - Our hypothesis: $T(n) \geq cn \lg n \qquad \forall n > n_0$ for some c, n_0
- Our Inductive Hypothesis is $T(m) \geq cm \lg m$ for all $m < n$.
  - So: $T(n) = 2T(n/2) + n$ (by definition)
- Using the inductive hypothesis, we get:

$$\begin{aligned}T(n) &\geq 2(cn/2 \lg n/2) + n\\ &= cn \lg(n/2) + n\\ &= cn(\lg n - \lg 2) + n\\ &= cn(\lg n - 1) + n\\ &= cn \lg n - cn + n\\ &= cn \lg n - (cn - n)\end{aligned}$$

- Recall we wish to prove $T(n) \geq cn \lg n$ so we must prove:
  - $cn \lg n - cn + n \geq cn \lg n$ or $n(1-c) > 0$
  - This is clearly true if $0 < c < 1$
- **QED**

Prove that *T(n) = 4 T(n/2) + n = O(n²)*

- Base case: $T(1) = \Theta(1)$
- Hypothesis: $T(n) \leq cn^2$
- $T(n) = 4T(n/2) + n \leq 4(c(n/2)^2 + n = cn^2 - (-n)$
- **Oops!** The "extra" is -n which cannot be greater than zero!
- Try a stronger hypothesis: $T(m) \leq cm^2 - cm \qquad \forall m < n$
- Then
  $T(n) = 4T(n/2) + n \leq 4(c(n/2)^2 - cn/2) + n = cn^2 - 2cn + n = cn^2 - cn - (c-1)n$
- Now "extra" is $n(c-1)$.
- If $c \geq 1$the "extra" is non-negative and the hypothesis is proven. **QED**
- (See also the example $T(n) = T(\lceil n/2 \rceil) + T(\lfloor n/2 \rfloor) + 1$ on page 85 of the MIT text book.)

Prove that *T(n) = 2T(n/4) +T(n/2) + n = O(n* log *n)*

- Inductive hypothesis: $T(m) \leq cm \lg m$ for all $m < n$.
- Using this hypothesis, we get:
-

$$
\begin{aligned}
T(n) &= 2T(n/4) + T(n/2) + n \\
T(n) &\leq 2(cn/4 \lg n/4) + c(n/2)\lg n/2 + n \\
&= \frac{cn}{2}\lg(n/4) + \frac{cn}{2}\lg n/2 + n \\
&= \frac{cn}{2}(\lg n - \lg 4) + \frac{cn}{2}(\lg n - \lg 2) + n \\
&= \frac{cn}{2}(\lg n - 2) + \frac{cn}{2}(\lg n - 1) + n \\
&= cn\lg n - 3cn/2 + n \\
&= cn\lg n - n(3c/2 - 1) \\
&= hypothesis - extra
\end{aligned}
$$

- The "extra" must be non-negative. This is assured for all *n* if $c \geq 2/3$.
- **QED**

Solving recurrences using recursion tree: more examples

T(n) = 2T(n/2) + 1

- This is similar to the merge sort recurrence (*T(n) = 2T(n) + 1)*.
- Except, only a constant (1) is added as the non-recursive part.
- Drawing a recursion tree, we would get the non-recursive part totals for each row as 1, 2, 4, 8, 16, etc.

- The depth of the tree would be $m = \lg n$, so the total would be
  $$\sum_{i=0}^{i=m=\lg n} 2^i = 2^{m+1} - 1 = 2n - 1$$
- Yes, this is an exact solution, but let's prove rigorously that $T(n) = \Theta(n)$
- First, let's prove that $T(n) = O(n)$, i.e. that $T(n) < cn$
  - True for base case: i.e. assume $T(1) \leq c$
  - Inductive hypothesis: $T(n) \leq cn$
  - Now prove that $T(2n) \leq c2n$
  - By definition: $T(2n) = 2T(n) + 1$
  - Substituting our inductive hypothesis, we must prove $T(2n) \leq 2cn + 1$
  - Oops!
  - Try a *stronger* inductive hypothesis: $T(n) = cn - k$
  - Now we need prove (for suitable constants) that $T(2n) \leq 2(cn - k) + 1 = 2cn - (k - 1)$

Exercises

1. Prove that $T(n) = 4T(n/2) + n = O(n^3)$.
2. Prove that $T(n) = 4T(n/2) + n = \Omega(n)$.

Questions

1. Draw a recursion tree for $T(n) = \sqrt{n}T(\sqrt{n}) + n$ for $n \geq 2$ and determine its $\Theta()$ complexity. (Hint: assume $n = 2^{2^m}$.)

Suggested problems

CLRS: 3-1, 3-2 (but not small-o and small-omega), 3-3, 3-4, 4-1, 4-3

My book: 4.1, 4.2, 4.3, 4.4, 4.7

References (text book and online)

- *CLRS:* Chapter 4.3
- kclowes book: Chapter 4

Appendix: Some Basic Math

$$\log_a x = \frac{\log_b x}{\log_b a}$$

$$a^{\log_a n} = n$$

$$a^{\log_b n} = n^{\log_b a}$$

$$\ln n < \sum_{i=1}^{n} 1/i = H_n < \ln n + 1 \textbf{ or } H_n = \Theta(\log n)$$

$$\sum_{i=1}^{n} i = n(n+1)/2$$

$$\sum_{i=0}^{n} r^i = \frac{r^{n+1}-1}{r-1} = \frac{1-r^{n+1}}{1-r}$$

$$\log n! = \Theta(n \log n)$$