A. V. Gerbessiotis — CIS 435
January 5, 2006 — Spring 2006
Brief Notes on Asymptotic Notation and Recurrences — Handout 4

A1. $f(n) = o(g(n))$, iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$.
A2. $f(n) = \omega(g(n))$, iff $\lim_{n\to\infty} \frac{f(n)}{g(n)} = \infty$.
A3. If $f(n) = o(g(n))$, then $f(n) = O(g(n))$.
A4. If $f(n) = \omega(g(n))$, then $f(n) = \Omega(g(n))$.
A5. If $f(n) = \Theta(g(n))$, iff $f(n) = \Omega(g(n))$ and $f(n) = O(g(n))$.
B1. $f(n) = \Theta(g(n))$ iff $\exists$ positive constants $c_1, c_2, n_0 : \; 0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n) \;\; \forall\, n \geq n_0$.
B2. $f(n) = \Omega(g(n))$ iff $\exists$ positive constants $c_1, n_0 : \; 0 \leq c_1 g(n) \leq f(n) \;\; \forall\, n \geq n_0$.
B3. $f(n) = O(g(n))$ iff $\exists$ positive constants $c_2, n_0 : \; 0 \leq f(n) \leq c_2 g(n) \;\; \forall\, n \geq n_0$.

Master Method. $T(n) = aT(n/b) + f(n)$, such that $a \geq 1$, $b > 1$.

M1 If $f(n) = O(n^{\lg_b a - \epsilon})$ for some constant $\epsilon > 0$, then $T(n) = \Theta(n^{\lg_b a})$.

M2 If $f(n) = \Theta(n^{\lg_b a})$, then $T(n) = \Theta(n^{\lg_b a} \lg n)$.

M3 If $f(n) = \Omega(n^{\lg_b a + \epsilon})$ for some constant $\epsilon > 0$, and if $af(n/b) \leq cf(n)$ for some constant $c < 1$ and for large n, then $T(n) = \Theta(f(n))$.

1 Short Notes on Asymptotic Notation

- **NEVER FORGET THAT** $O(f(n))$ is not a function, it is a SET. Therefore a relation of the form $1 \leq O(n)$ is nonsense as you compare a constant function (1) to a set using $\leq$ a symbol that we use to compare numbers.
- ω vs o? Take the limit of the two functions $f(n)/g(n)$. If the limit is 0, then $f(n) = o(g(n))$. If it is ∞, then $f(n) = \omega(g(n))$.
- If little o is true then O is also true; if $f(n) = o(g(n))$, then $f(n) = O(g(n))$. Prove it.
- If little ω is true then Ω is also true; if $f(n) = \omega(g(n)$, then $f(n) = \Omega(g(n))$. Prove it.

We review the following relationships (n is positive).

- Remember $2^{\lg n} = n$ and $a^{\lg n} = n^{\lg a}$ and $n^{1/\lg n} = 2$.
- $n = \omega(\lg n)$, $n = \omega(\lg^2 n) = \omega((\lg n)^2)$. Note that $\lg^2 n$ means $(\lg n)^2$.
- In general $n^k > \lg^l n$ for any constant k, l and large enough n. It is also true that $n^k = \omega(\lg^l n)$.
- In general $2^n > n^k$ for any constant k and large enough n. It is also true that $2^n = \omega(n^k)$.
- $n! > n^k$ for any constant k and large enough n.
- $n! \approx (n/e)^n \sqrt{2\pi n}$ (Stirling's approximation formula for the factorial). It is also true that $\lg(n!) = O(n \lg n)$.
- Again note that $n^{1/\lg n} = 2$. $n^{1/\lg n}$ seems **to grow fast** but it does not! It is a constant, the constant 2.

Example 1 *Which of $a_0 + a_1 n + a_2 n^2 + a_3 n^3$ and n^2 is asymptotically larger, where $a_i > 0$ for all i?*

Proof. Consider $a_0 + a_1 n + a_2 n^2 + a_3 n^3$. As all a_i are positive, then $a_0 > 0$ and $a_1 n > 0$ and $a_2 n^2 > 0$ and thus $a_0 + a_1 n + a_2 n^2 + a_3 n^3 > a_3 n^3$.

Hint. When we intend to prove $f(n) = \Omega(g(n))$, it sometimes helps to find a lower bound $h(n)$ for $f(n)$ ie one such that $f(n) \geq h(n)$ and then show that $h(n) = \Omega(g(n))$. In our case a lower bound for $a_0 + a_1 n + a_2 n^2 + a_3 n^3$ is $a_3 n^3$.

We now show that our lower bound $a_3 n^3$ is $\Omega(n^2)$. As $a_3 > 0$, it is obvious that $a_3 n^3 > 1 \cdot n^2$ for any $n > 1/a_3$ (note that $a_3 > 0$ DOES NOT MEAN THAT $a_3 > 1$, as a_3 is real and not necessarily an integer).

Therefore for $c = 1$ and $n_0 = 1/a_3$ we have shown that $a_0 + a_1 n + a_2 n^2 + a_3 n^3 = \Omega(n^2)$. ∎

Example 2 *Which of the two functions is asymptotically larger $a_0 + a_1 n + a_2 n^2 + a_3 n^3$ or n^4, where $a_i > 0$ for all i?*

Hint. When we intend to prove $f(n) = O(g(n))$, as is the case here, it sometimes helps to find an upper bound $h(n)$ for $f(n)$ ie one such that $f(n) \leq h(n)$ and then show that $h(n) = O(g(n))$.

Since in $a_0 + a_1 n + a_2 n^2 + a_3 n^3$ all a_i are positive we take the maximum of all a_i and we call it A. Then we have that $a_i < A$ for all i. Also, $An^i < An^3$ for $i \leq 3$. Then

$$a_0 + a_1 n + a_2 n^2 + a_3 n^3 \leq A + An + An^2 + An^3 \leq An^3 + An^3 + An^3 + An^3 = 4An^3$$

Finally $4An^3 \leq n^4$ for all $n > 4A$.

We have shown that

$$a_0 + a_1 n + a_2 n^2 + a_3 n^3 \leq 4An^3 \leq 1 \cdot n^4$$

for all $n \geq n_0 = 4A$, where A is the maximum of a_0, a_1, a_2, a_3. As all a_i are constant, so is $4A$. Therefore the constants in the O definition are $c = 1$ and $n_0 = 4A$, where $A = \max\{a_0, a_1, a_2, a_3\}$. ∎

Examples on Recurrences

Example 3 *Solve the recurrence $T(n) = 2T(n/2) + n$ using the substitution (a.k.a. guess-and-check) method. Implicit assumption: $T(n)$ is nonnegative for all n.*

Proof. This recurrence has already been solved in Handout 3 (strong induction) where a very tight upper

bound for $T(n)$ was given. We observe that no boundary condition is given; we can thus assume T(constant) = some-constant.

Guess a solution. We guess $T(n) \leq cn \lg n$ for all $n_0 \leq n$ where c, n_0 are positive constants. This is equivalent

to showing that $T(n) = O(n \lg n)$. We call this the predicate $P(n)$. We are going to show that $P(n)$ is true for all $n \geq n_0$, ie. we are going to show that the guessed solution is the solution to the recurrence. We shall prove our claim by using strong induction. In the inductive proof we shall delay the proof of the BASE CASE until the very end.

Base Case. Proof at the end (let us assume that base case is true for the remainder).

Induction Hypothesis. Assume $P(n_0) \wedge P(n_0+1) \wedge \ldots P(n-1)$ **are true.** We assume that the guessed solution is true for all integer values less than n ie that for all $i \leq n-1$ we have that

$$T(i) \leq ci \lg i \text{ for all } n_0 \leq i < n-1,$$

(Here we renamed the variable in $T(.)$ to i from n to avoid confusion).

Inductive Step: Show that $P(n_0) \wedge P(n_0+1) \wedge \ldots P(n-1) \Rightarrow P(n)$**.**

In the inductive step we show the correctness of our guess for $i = n$ i.e. for the next largest value of i (by the induction hypothesis the inequality is true for $i \leq n-1$).

$$T(i) \leq ci \lg i \text{ for } i = n$$

which is equivalent to showing that

$$T(n) \leq cn \lg n$$

Step 1.

As $n/2 < n$, the induction hypothesis applies to $i = n/2$. Therefore,

$$T(n/2) \leq c(n/2) \lg (n/2) \qquad (1)$$

Step 2.

In order to prove the inductive step we use the only piece of information that we have available for $T(n)$. **This is the recurrence relation**. The inequality below is a result of the induction hypothesis.

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

Step 3.

In order to show that $T(n) \leq cn \lg n$, the last expression $cn \lg n - cn + n$ must be less than or equal to $cn \lg n$. This is so provided that $c \geq 1$. (a **condition on c is thus established**).

Therefore for $c \geq 1$ we have that

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

We have thus shown the inductive step. This completes the induction, i.e. we have proved that $T(n) \leq cn \lg n$ for any $c \geq 1$, AS LONG AS THE BASE CASE IS TRUE. We show now the base case.

Base Case Proof. As $\lg 0$ is not defined, we don't define a base case for $T(0)$. The base case is defined for an $n = n_0$, where n_0 is some constant. For $n = n_0$, the recurrence $T(n_0) = 2T(n_0/2) + n_0$ and the claim we want to establish $T(n_0) \leq cn_0 \lg n_0$ must both hold.

We first attempt to prove the base case for an $n_0 = 1$. To show that $T(1) \leq c \cdot 1 \cdot \lg 1$, as $c \cdot 1 \cdot \lg 1 = 0$, we must have that $T(1) \leq 0$. As $T(1) \geq 0$ ($T(n)$ is assumed to be non-negative), then we must have $T(1) = 0$ as the boundary condition.

Conclusion 1 (Case $T(1) = 0$)

A solution to the recurrence $T(n) = 2T(n/2) + n$, $T(1) = 0$ is $T(n) \leq cn \lg n$ for any $c \geq 1$ and $n \geq 1$.

What if the boundary condition is $T(1) = 3$ and not $T(1) = 0$?

Then, from the previous discussion $T(1) = 3$ and $T(1) \leq 0$ must BOTH hold, a contradiction. This means that a base case cannot be established for $n_0 = 1$. The next largest value for n_0 is 2.

We thus try the next available integer value $n = 2$. From the recurrence we have that $T(2) = 2T(1) + 2 = 2 \cdot 3 + 2 = 8$, as $T(1) = 3$. Then $T(2) < c \cdot 2 \cdot \lg 2 = 2c$ is true provided that $8 \leq 2c$, i.e. $c \geq 4$.

Conclusion 2 (Case $T(1) = 3$)

A solution to the recurrence $T(n) = 2T(n/2) + n$, $T(1) = 3$ is $T(n) \leq cn \lg n$ for any $c \geq 4$ and $n \geq 2$.

■

We now solve the same recurrence using the iteration (a.k.a recursion tree) method, a frequently occurring exam problem. We change the boundary condition from the one used in class.

Iteration Method (Recursion-tree method).

Example 4 *Solve the recurrence* $T(n) = 2T(n/2) + n$, $T(2) = 5$ *using the iteration/recursion tree method.*

Proof.

We rename variables in the recurrence relation substituting i for n (it doesn't matter whether we have i or n, but the discussion below will be become less confusing).

$$T(i) = 2T(i/2) + i$$

Substituting $n/2$ for i we get.

$$T(n/2) = 2T((n/2)/2) + n/2 = 2T(n/2^2) + n/2$$

Substituting $n/4$ for i we get.

$$T(n/2^2) = T(n/4) = 2T((n/4)/2) + n/4 = 2T(n/2^3) + n/4$$

Similarly we can substitute $n/2^3, n/2^4, \ldots, n/2^i, \ldots, n/2^{lgn} = 1$ for i and get similarly stated recurrences.

We use all the derived recurrences to expand $T(n)$ below. The objective is to sum the equalities and observe in the expansion sequence some symmetries and repetitions for the purpose of combining as many terms as possible to derive a closed form solution for $T(n)$.

$$\begin{aligned}
T(n) &= 2T(n/2) + n \\
&= 2(2T(n/2^2) + n/2) + n \\
&= 2^2T(n/2^2) + 2 \cdot n/2 + n \\
&= 2^2(2T(n/2^3) + n/2^2) + 2 \cdot n/2 + n \\
&= 2^3T(n/2^3) + 2^2(n/2^2) + 2 \cdot n/2 + n \\
&= 2^3T(n/2^3) + n + n + n \\
&= 2^3T(n/2^3) + 3n \\
&= \ldots \\
&= 2^iT(n/2^i) + i \cdot n
\end{aligned}$$

From the boundary condition for $m = 2$, which is $T(2) = 5$, we decide when to stop the expansion. The expansion ends so that $n/2^i = 2$ and then $T(n/2^i) = T(2) = 5$. We solve for i by taking logarithms base two of both sides. Then we get that $\lg n - i = 1$ ie $i = \lg n - 1$.

Then, for $i = \lg n - 1$, we get that.

$$\begin{aligned}
T(n) &= 2T(n/2) + n \\
&= 2^i T(n/2^i) + i \cdot n \\
&= 2^{\lg n - 1} T(n/2^{\lg n - 1}) + (\lg n - 1) \cdot n \\
&= (n/2)T(2) + (\lg n - 1) \cdot n \\
&= (n/2)5 + (\lg n - 1) \cdot n \\
&= (3n/2) + n \lg n
\end{aligned}$$

Thus we have obtained the following solution to the recurrence: $T(n) = 3n/2 + n \lg n$.

If you are not sure about the correctness of your calculations, you can do the following to verify the correctness of your derivations.

Checking the solution (THIS IS NOT PART OF A SOLUTION).

How can we be sure of the correctness of this solution? It suffices to show that the derived solution (a) validates the recurrence, and (b) is consistent with $T(2)$.

We start with (b).

$$T(2) = 3 \cdot 2/2 + 2 \lg 2 = 3 + 2 = 5 = T(2).$$

It is obvious that the boundary condition is consistent with the solution.

We proceed to showing (a). We obtained the solution

$$T(n) = 3n/2 + n \lg n.$$

Substituting $n/2$ for n we get that

$$T(n/2) = 3n/4 + (n/2) \lg n/2 = 3n/4 + (n/2)(\lg n - 1) = n/4 + (n/2) \lg n.$$

We start from the right hand side of the recurrence using the preceding equality.

$$2T(n/2) + n = 2(n/4 + n/2 \lg n) + n = 3n/2 + n \lg n.$$

The last term is $T(n)$. We have thus proved that for $T(n) = 3n/2 + n \lg n$, we have that $T(n) = 2T(n/2) + n$, ie our solution satisfies both the recurrence and the boundary condition, ie it is indeed a solution to the recurrence. ■.

Example 5 *Solve the recurrence $T(n) = 8T(n/2) + n$ using the iteration/recursion tree method. Assume that $T(1) = 5$.*

Proof.

$$\begin{aligned}
T(n) &= 8T(n/2) + n \\
&= 8(8T(n/2^2) + n/2) + n \\
&= 8^2T(n/2^2) + 8n/2 + n \\
&= 8^2T(n/2^2) + (8/2)^1 n + (8/2)^0 n \\
&= 8^2(8T(n/2^3) + n/2^2) + (8/2)^1 n + (8/2)^0 n \\
&= 8^3T(n/2^3) + 8^2 n/2^2 + (8/2)^1 n + (8/2)^0 n \\
&= 8^3T(n/2^3) + (8/2)^2 n + (8/2)^1 n + (8/2)^0 n \\
&= \ldots \\
&= 8^iT(n/2^i) + (8/2)^{i-1} n + \ldots + (8/2)^1 n + (8/2)^0 n
\end{aligned}$$

Again the boundary case is $T(1) = 5$. We set $n/2^i = 1$, ie $i = \lg n$. Then for $i = \lg n$, $T(n/2^i) = T(1) = 5$. We therefore get for $i = \lg n$.

$$\begin{aligned}
T(n) &= 8T(n/2) + n \\
&= 8^iT(n/2^i) + (8/2)^{i-1} n + \ldots + (8/2)^1 n + (8/2)^0 n \\
&= 8^{\lg n}T(n/2^{\lg n}) + \frac{(8/2)^{\lg n} - 1}{8/2 - 1} \cdot n \\
&= 2^{3\lg n}T(1) + \frac{(4)^{\lg n} - 1}{3} \cdot n \\
&= n^3T(1) + \frac{(4)^{\lg n} - 1}{3} \cdot n \\
&= 5n^3 + \frac{n^2 - 1}{3} n
\end{aligned}$$

To verify our calculations we observe that $T(1) = 5 + (1 - 1)/3 = 5$ and

$$\begin{aligned}
T(n) &= 8T(n/2) + n \\
&= 8(5(n/2)^3 + \frac{(n/2)^2 - 1}{3} n) + n \\
&= 5n^3 + \frac{n^2 - 1}{3} n \\
&= T(n),
\end{aligned}$$

ie the recurrence is verified.

Additional problems and exercices

Exercise 0.

Do the Exercides of the textbook for the chapters/sections covered in class. The more you do of them the more you practice.

Exercise 1.

Calculate the following sum for any $x \neq 1$

$$x + 2x^2 + 3x^3 + \ldots + nx^n = \sum_{i=1}^{n} ix^i.$$

(*Hint:* Consult the appendix (Appendix A) on page 1060.

Exercise 2.

Show that

$$\sum_{i=1}^{n} i^2 = \Theta(n^3).$$

What are the values of c_1, c_2 and n_0? Justify your answer.

Exercise 3.

TRUE or FALSE?

1. $\lg(n!) = O(n^2)$.
2. $n + \sqrt{n} = O(n^2)$.
3. $n^2 + \sqrt{n} = O(n^2)$.
4. $n^3 + 2\sqrt{n} = O(n^2)$.
5. $1/n^3 = O(\lg n)$.
6. $n^2\ sin^2(n) = \Theta(n^2)$. ($sin$ is the well-known trigonometric function).

Exercise 4.

Prove the following.

1. $(n - 10)^2 = \Theta(n^2)$.
2. $n^4 + 10n^3 + 100n^2 + 1890n + 98000 = \Omega(n^4)$.
3. $n^4 + 10n^3 + 100n^2 + 1890n + 98000 = \Omega(n^2)$.
4. $n^4 - 10n^3 - 100n^2 - 1890n + 100000 = O(n^4)$.
5. $n^2 - 20n - 20 = \Omega(n)$.

6. $n^2 + 20n = O(n^2)$.

Exercise 5.

Solve the following recurrences. You may assume $T(1) = \Theta(1)$, where necessary. Make your bounds as tight as possible. Use asymptotic notation to express your answers. Justify your answers.

a. $T(n) = 2T(n/8) + n$
b. $T(n) = 9T(n/3) + n \lg^2 n$
c. $T(n) = 3T(n/9) + n^2$
d. $T(n) = 2T(n/4) + \sqrt{n}$
e. $T(n) = 4T(n/2) + n$.
f. $T(n) = 2T(n/16) + n^{1/4}$.
g. $T(n) = T(n/2) + 1$, $T(1) = 1$.

Exercise 6.

Solve the following recurrences. Make your bounds as tight as possible. Use asymptotic notation to express your answers. Justify your answers.

a. $T(n) = T(n/8) + T(7n/8) + n \lg n \quad T(1) = 100$
b. $T(n) = T(n/5) + T(3n/4) + 10n$, $T(1) = 20$.

Exercise 7.

Find an asymptotically tight bound for the following recurrence. You may assume $T(1) = \Theta(1)$. Justify your answer.

$T(n) = T(n/8) + T(3n/4) + 8n$.

Exercise 8.

Solve exactly using the iteration method the following recurrence. You may assume that n is a power of 3, ie $n = 3^k$.

$$T(n) = 3T(n/3) + n, \text{ where } T(1) = 1.$$

Solutions/Hints to selected problems.

Exercise 2.

$\Sigma_{i=1}^{n} i^2 = n(n+1)(2n+1)/6$. One can show this by induction, or refer to Handout 3, pages 3 and 4, or use the technique of page 7. We thus focus ourselves on $n(n+1)(2n+1)/6$.

Case 1.

$$n(n+1)(2n+1)/6 \leq n(n+1)(2n+1) \leq n(n+n)(2n+n) = n(2n)(3n) = 6n^3$$

This shows that $n(n+1)(2n+1)/6$ is $O(n^3)$.

Case 2.

$$n(n+1)(2n+1)/6 \geq n(n)(2n)/6 \geq n^3/3$$

This shows that $n(n+1)(2n+1)/6$ is $\Omega(n^3)$.

The combination of Cases 1 and 2 shows the result.

Exercise 3.

1. $n!$ by Sterling's approximation formula is $n! \approx (n/e)^n$. Therefore $\lg n! = \Theta(n \lg n)$ and the results follows (i.e. answer is TRUE).

Exercise 4.

1. $(n-10)^2 = n^2 - 2 \cdot 10n + 10^2 = n^2 - 20n + 100$.

We first show the $O(.)$ part.

$$n^2 - 20n + 100 \leq n^2 + 0 + 100n^2 = 101n^2$$

We have just shown that $(n-10)^2 \leq 101n^2$ for all $n \geq 0$, ie. established $c_2 = 101$ and $n_2 = 0$.

We then show the $\Omega(.)$ part.

$$n^2 - 20n + 100 \geq n^2 - 20n \geq n^2/2$$

For the second inequality to be true $n^2 - 20n \geq n^2/2$ i.e. $n^2/2 \geq 20n$, i.e. $n \geq 40$. I.e. We have just shown that $(n-10)^2 \geq n^2/2$ for all $n \geq 40$, ie. established $c_1 = 1/2$ and $n_1 = 40$.

Therefore for the $\Theta(.)$ definition the constants are $c_1 = 1/2$, $c_2 = 101$ and $n_0 = \max\{n_1, n_2\} = 40$.

Exercise 6.

a. Hint. Show that $T(n) = O(n \lg^2 n)$. Can you show $T(n) = \Theta(n \lg^2 n)$?

b. Hint. Show that $T(n) = \Theta(n)$.

Exercise 7.

See 6b.

Exercise 8.
Similar to $T(n) = 2T(n/2) + n$, but do not forget that $\lg n$ was $\lg_2 n$ in the former. ■