Section II.5: Growth Rates of Functions

One of the most important problems in computer science is to get the **best** measure of the growth rates of algorithms, **best** being those algorithms whose run times grow the slowest as a function of the size of their input. Efficiency can mean survival of a company. For example, a sort of measure O(n^2) on a database of millions of customers may take several days to run, whereas one of measure O($n \cdot \log n$) may take only a few minutes! See Table 9.3.1, Epp p. 497 for comparisons of run times.

However, the big O estimate, defined and given in Epp, section 9.2, does **not** necessarily give the best measure of the growth rate of a function. One can say that the growth rate of a sequential search is O(n^2), but one knows the number of comparisons is approximately proportional to n, n the number of input items (see the notes on Expected Values and Applications, Section I.2). We would like to say that sequential search is O(n) (it is), but the notion of big O is not precise enough. Therefore, in this section we define theta Θ notation to more precisely measure the growth rate of functions and big omega Ω notation. The most important of these is Θ**.** We also define o and ω notation.

Definition II.5.1: Let f and g be real valued functions, X a real number.
$f(X) = \Omega(g(X))$, i.e., $f = \Omega(g)$, if and only if there are positive real constants C and X_0 such that $| f(X) | \geq C \cdot | g(X) |$ for all $X > X_0$.

Note: $f(X) = \Omega(g(X))$ if and only if $g(X) = O(f(X))$.

Example II.5.1: Show that **(a)** $2X^2 = \Omega(X^2 + 1)$, **(b)** $X^2 = \Omega(2X)$, and
(c) $2X \neq \Omega(X^2)$.

Solution:
(a) We wish to find C and X_0. that satisfy the conditions of the definition of Ω. Let $C = 1$ and solve $2X^2 \geq (X^2 + 1)$. This holds if $X \geq 1$. Let $X_0 = 1$. Then $C = 1$ and $X_0 = 1$ guarantee that $2X^2 \geq C(X^2 + 1)$ for all $X > X_0$.

(b) We wish to find C and X_0 that satisfy the conditions of the definition of Ω.
Let $C = 1$. Solve $X^2 \geq 2X$. This holds when $X^2 - 2X \geq 0$, or when $X \geq 2$. $C = 1$ and $X_0 = 2$ work.

(c) This is equivalent to showing $X^2 \neq O(2X)$. We assume $X^2 = O(2X)$ and get a contradiction. Let there exist positive reals C and X_0 such that $X^2 \leq C(2X)$ for all $X > X_0$. Divide the inequality by X. One gets $X \leq 2C$, which means X is bounded above. However, $X^2 \leq C(2X)$ must occur for all $X > X_0$, X_0 a constant. This implies X is unbounded above. X bounded and X unbounded give a contradiction. Therefore, our assumption that $X^2 = O(2X)$ cannot be true.

Definition II.5.2: Let f and g be real valued functions, X a real number. $f(X) = \Theta(g(X))$, i. e., $f = \Theta(g)$ if and only if $f(X) = O(g(X))$ and $g(X) = O(f(X)$. Equivalently, $f(X) = \Theta(g(X))$ if and only if $f(X) = O(g(X)$ and $f(X) = \Omega(g(X))$ or, if and only if $f(X) = \Omega(g(X)$ and $g(X) = \Omega(f(X))$.

Note: When we say "get the best big O estimate for function f", we are really saying "find g such that $f(X) = \Theta(g(X))$", i. e., we are finding the smallest growth rate for f, or we are finding a function g "simpler" than f which has the same growth rate as f. The function g is ideally a function with one term and coefficient 1. The primary application of these ideas is in finding the run time for an algorithm, meaning the **slowest** run time.

Lemma II.5.1: $f(X) = \Theta(g(X)$ if and only if there are positive constants C_1, C_2, and X_0 such that for all $X > X_0$, $C_1 \mid g(X) \mid \leq \mid f(X) \mid \leq C_2 \mid g(X) \mid$.

Proof:

(a) We must show that, if $f(X) = \Theta(g(X)$, then there are positive constants C_1, C_2, and X_0 such that for all $X > X_0$, $C_1 \mid g(X) \mid \leq \mid f(X) \mid \leq C_2 \mid g(X) \mid$.

Proof of (a): We are given that **(1)** $f(X) = O(g(X)$ and **(2)** $f(X) = \Omega(X)$. **(1)** implies that there are positive constants X_2 and C_2 such that $\mid f(X) \mid \leq C_2 \mid g(X) \mid$ for all $X > X_2$. **(2)** implies that there are positive constants X_1 and C_1 such that $C_1 \mid g(X) \mid \leq \mid f(X) \mid$ for all $X > X_1$.

Let $X_0 = max \{ X_1, X_2 \}$. If $X > X_0$, then $X > X_1$ and $X > X_2$. Combining the results of **(1)** and **(2)**, we have, for all $X > X_0$, $C_1 \mid g(X) \mid \leq \mid f(X) \mid \leq C_2 \mid g(X) \mid$.

(b) To show that, if there are positive constants C_1, C_2, and X_0 such that, for all $X > X_0$, $C_1 \mid g(X) \mid \leq \mid f(X) \mid \leq C_2 \mid g(X) \mid$, then $f(X) = \Theta(g(X)$, we must show **(1)** $f(X) = O(g(X)$, and **(2)** $f(X) = \Omega(X)$.

Proof of (b):

(1) holds since there are constants C_2 and X_0 such that for all $X > X_0$,
$\mid f(X) \mid \leq C_2 \mid g(X) \mid$

(**2**) holds since there are constants C_1 and X_0 such that for all $X > X_0$,
$C_1 \mid g(X) \mid \leq \mid f(X) \mid$.

Example II.5.2:
(a) In Example II.5.1 (c), $X^2 \neq \Theta(2X)$ since $X^2 = \Omega(2X)$ and $2X \neq \Omega(X^2)$.

(b) Show $2X^2 = \Theta(X^2 + 1)$.

Solution: In example II.5.1, it was shown that $2X^2 = \Omega(X^2 + 1)$. It suffices to show that $2X^2 = O(X^2 + 1)$. Let $C = 2$ and X_0 be any real number. Since $2X^2 \leq 2(X^2 + 1)$ for any X, we know $C = 2$ and any real X_0 satisfy the conditions in the definition of big O.

The examples above show that the growth rates of X^2 and $2X$ are not the same. X^2 grows faster than $2X$, equivalently $2X$ grows more slowly than X^2. If one doubles X, then $2X$ is doubled, but X^2 is multiplied by 4. However, $2X^2$ and $X^2 + 1$ have the same growth rate. If X is doubled, both of these are multiplied by a factor close to 4.

Classes of Growth Rates

Functions can be put into disjoint classes where the growth rates of all functions in a class are the same. If C is such a class, we can define C as the set of all functions f where $f = \Theta(g)$, g a particular function in C. These classes form an ascending chain of growth rates. Some of these classes are listed below. We define <' among functions to mean f <' g if and only if $f = O(g)$ but $g \neq O(f)$. Using n (a positive integer) as a variable rather than X (since input size is an integer) and $c > 2$ a constant, we can list some of the classes of growth rates as

$$1 <' \log(n) <' \ldots <' n <' n \cdot \log(n) <' \ldots <' n^2 <' \ldots <' n^c <' \ldots <' 2^n <' \ldots <' c^n <' \ldots <' n! <' \ldots <' n^n <' \ldots$$

where … indicates there are classes between these two classes.

Note: <' is **transitive**, i.e., if $f <' g$ and $g <' h$, then $f <' h$. We can define $f =' g$ to mean f and g have the same growth rate and $f >' g$ as g < f. Also $f \leq' g$ means either $f =' g$ or $f <' g$.

Definition II.5.3: Little o and lower case ω.
For completeness we define $f = o(g)$ to mean $f = O(g)$ and $g \neq O(f)$. Also $f = \omega(g)$ means $f = \Omega(g)$ but $g \neq \Omega(f)$.

Example II.5.3: From the examples above, one can see that $X^2 = \omega(2X)$ and $2X = o(X^2)$. However $(2X^2) \neq \omega(X^2 + 1)$ and $(2X^2) \neq o(X^2 + 1)$.

There is a natural correspondence between definitions involving growth rates of functions and the ordinary ordering relations. The table below illustrates this relationship.

Growth rates:	o	O	Θ	Ω	ω
Ordering relations	$<$	$\leq$	$=$	$\geq$	$>$

Finding theta estimates for growth rates, with coefficient 1.

In the analysis of the growth rates of algorithms, what matters is the term with the largest growth rate (dominant term). Because of C in the definition of growth rate, one can consider the coefficient to be 1. The below chart gives common terminology for growth rates from functions in various growth rate classes and examples of algorithms in each (some to be analyzed later in this course).

Class	Terminology	Example
1	Constant growth	Finding midpoint of an array
$\log(n)$	Logarithmic growth	Binary search
n	Linear growth	Linear Search
$n \cdot \log(n)$		Merge Sort
n^2	Quadratic growth	Insertion Sort
n^3	Cubic growth	Seeing if an element appears 3 times in a list
n^c	Polynomial growth	The two above
c^n	Exponential growth	Towers of Hanoi
$n!$	Factorial growth	Traveling salesman problem

In view of Theorem II.5.8, the growth rate of $\log_c n$ is the same irrespective of the value of constant $c > 1$. So we shall use $\log(n)$ to mean $\log_c n$ for any constant $c > 0$. The most important base in growth rates is 2, and $\lg(n)$ is common notation for $\log_2 n$.

An algorithm with growth rate equal to or larger than exponential growth is called **intractable** because, for even moderate input size, its run time is impractically long (Epp. p.530); also see table 9.3.1, Epp. A major concern in computer science is to find **tractable** (run times polynomial time or better) algorithms to solve the problem. If a problem has no tractable algorithm for its solution, then the objective is to find a tractable algorithm to approximate its solution.

The next few theorems enable us to get Θ estimates, single term with coefficient 1, for functions with several terms.

Theorem II.5.1: If $x > 1$ and $r < s$, r and s positive and rational, then $x^r = o(x^s)$.

Outline of Proof: Show **(a)** $x^r = O(x^s)$ but **(b)** $x^s \neq O(x^r)$.
In (a) take $C = 1$ and $X_0 = 1$. The result follows from the definition of big O.
In (b) assume $x^s = O(x^r)$. Using the fact that s-$r > 0$, and taking $X_0 = 1$ in the definition of big O, one can get a contradiction. (Exercise).

Theorem II.5.2: (Adaptation of Epp, Theorem 9.2.1) If f is a polynomial of degree n, then $f = \Theta(x^n)$.

The next two theorems are very important since they enable us to get estimates with coefficient 1. Proofs of the corresponding big O statements are in Rosen, p. 80.

Theorem II.5.3: If $f_1 = \Theta(g_1)$ and $f_2 = \Theta(g_2)$ then $f_1 + f_2 = \Theta(max\{g_1, g_2\})$ where $max\{g_1, g_2\}$ is in the class with the larger growth rate of g_1 or g_2.

Theorem II.5.4: If $f_1 = \Theta(g_1)$ and $f_2 = \Theta(g_2)$ then $f_1 \cdot f_2 = \Theta(g_1 \cdot g_2)$.
Proof: One method is to show this result holds for both the Ω case and the big O case. Here we do the proof in the Ω case and leave the big O case as an exercise.

Since $f_1 = \Omega(g_1)$, we are given that there are positive constants X_1 and C_1 such that $C_1 \, |g_1(X)| \leq |f_1(X)|$ for all $X > X_1$.

Since $f_2 = \Omega(g_2)$, we are given that there are positive constants X_2 and C_2 such that $C_2 \, |g_2(X)| \leq |f_2(X)|$ for all $X > X_2$.$\leq |f_1(X) \cdot f_2(X)|$ for all $X > X_2$.

We must **show** that there are positive constants X_0 and C such that $C \, |g_1(X) \cdot g_2(X)| \leq |f_1(X) \cdot f_2(X)|$ for all $X > X_0$.

Let $X_0 = \max\{X_1, X_2\}$. Multiplying the above two inequalities together, and using an argument like that in proof of Lemma II.5.1, we get
$C_1 \, |g_1(X)| \, C_2 \, |g_2(X)| \leq |f_1(X) \cdot f_2(X)|$ for all $X > X_0$. Using the equation $|a \cdot b| = |a| \cdot |b|$, we get $C_1 \cdot C_2 \cdot |g_1(X) \cdot g_2(X)| \leq |f_1(X) \cdot f_2(X)|$.
Taking $C = C_1 \cdot C_2$ and $X_0 = \max\{X_1, X_2\}$, we have satisfied the definition of $f_1(X) \cdot f_2(X) = \Omega(g_1(X) \cdot g_2(X))$.

Example II.5.4: Get the Θ estimates, coefficient 1, for the functions
(a) $f(x) = (x^3 + x^2 \cdot \log(x)) \cdot (x \cdot \log(x))$
(b) $g(x) = 2^x + (x^5 - 100x^4 + 43)$.

Solution:
(a) Let $h(x) = (x^3 + x^2 \cdot \log(x))$. By the growth chart above, $x \cdot \log(x) <' x^2$. One can assume $x > 0$. Multiplying this inequality by x, one obtains $x^2 \cdot \log(x) <' x^3$. Therefore, $\max\{x^3, x^2 \cdot \log(x)\} = x^3$. By theorem II.5.3, $h(x) = \Theta(x^3)$. Applying theorem II.5.4, we get $f(x) = \Theta(x^3 \cdot x \cdot \log(x)) = \Theta(x^4 \cdot \log(x))$.

(b) Let $j(x) = x^5 - 100x^4 + 43$. By Theorem II.5.2, $j(x) = \Theta(x^5)$. From the growth rate table, $\max\{x^5, 2^x\} = 2^x$. Therefore, by Theorem II.5.3, $g(x) = \Theta(2^x)$.

One can prove rigorously that the table above really gives disjoint classes of growth rates and that the order is correct. For example, Theorem II.5.1 convinces us that $n <' n^2$. It remains to be shown that $\log(n) <' n$, $n^c <' a^n$, $a^n <' n!$, $n^a <' n^b$, and $a^n <' b^n$ when $0 < a < b$, a, b and $c > 0$ are constant. Some of these facts will be proved below. Proofs of others can be found in the literature or be tried as exercises.

A very important result that will be assumed without proof is

Theorem II.5.5: $f <' g$ if and only if $\lim_{x\to\infty} \frac{f(x)}{g(x)} = 0$.

Theorem II.5.6: $\log(x) <' x^c$ for any $c > 0$.

Proof: We use L'Hôpital's rule that $\lim_{x\to\infty} \frac{f(x)}{g(x)} = \lim_{x\to\infty} \frac{f'(x)}{g'(x)}$
whenever $\frac{f(x)}{g(x)}$ is of the form $\frac{0}{0}$ or $\frac{\infty}{\infty}$ and $\frac{f'(x)}{g'(x)}$ is **not** of the form $\frac{0}{0}$ or $\frac{\infty}{\infty}$.

$\mathrm{Log}(x)/x^c$ is of the form $\frac{\infty}{\infty}$. Without loss of generality, assume
$\log(x) = \log_e(x) = \ln(x)$. Then $\lim_{x\to\infty} \ln(x)/x^c = \lim_{x\to\infty} 1/(c \cdot x^{c-1} \cdot x) = \lim_{x\to\infty} 1/(c \cdot x^c) = 0$
since $c > 0$ and x gets arbitrarily large. By Theorem II.5.5, we have $\log(x) <' x^c$.

Corollary: $\log(x) <' x$.

Proof: Apply Theorem II.5.6 with $c = 1$.

Theorem II.5.7: a^n <' b^n when $0 < a < b$.

Proof: Exercise. Use the fact that $\lim_{x\to\infty} c^x = 0$ when $|c| < 1$.

Theorem II.5.8: $\log_a(x) = \Theta(\log_b(x))$ where a, b are positive constants larger than 1.

Proof: We show that **(a)** $\log_a(x) = O(\log_b(x))$ and **(b)** $\log_b(x) = O(\log_a(x))$. By the change of base formula, $\log_b(x) = \log_a(x)/\log_a(b)$. Using the definition of big O, let $C = |1/\log_a(b)|$ and x_0 be any positive number. Then $\log_a(x) = C \cdot \log_b(x)$ for any $x > x_0$. To show $\log_b(x) = O(\log_a(x))$, use a similar argument with the same x_0 and $C = |\log_a(b)|$.

Example II.5.5: Show that $n! <' n^n$.

Solution: We write out the factors of $n!$ and n^n and show $\lim_{n\to\infty} \frac{n!}{n^n} = 0$.

$\frac{n!}{n^n} = \left(\frac{1}{n}\right)\left(\frac{2}{n}\right)\cdots\left(\frac{n}{n}\right)$, and as $n \to \infty$, this product goes to zero since all factors but the last have absolute value less than 1, there are more and more of them, and the last factor is 1.

Exercises:

(1) Indicate true or false. If the result is true, verify by the formal definitions of O, Θ, or Ω, whichever applies. If it is false, show why it is false. For questions concerning o or ω, answer in terms of the definitions of O, Θ, or Ω, i.e., o is O and not Θ, and ω is Ω and not Θ. Let $f(x) = x^3$, $g(x) = x^2 + 6x$, and $h(x) = 2x^2$

(a) $f(x) = \Omega(h(x))$ **(b)** $h(x) = \Omega(g(x))$ **(c)** $h(x) = \Theta(g(x))$
(d) $h(x) = \Theta(f(x))$ **(e)** $f(x) = \omega(h(x))$ **(f)** $h(x) = o(f(x))$
(g) $g(x) = o(h(x))$ **(h)** $h(x) = \omega(g(x))$

(2) Give the Θ estimates, single term, coefficient 1, for the functions below. Use the theorems in this section to justify your work. Study examples II.5.4 (a) and (b).

(a) $f(n) = n! + (n \cdot \log(n)) \cdot (n^2 + 1)$
(b) $f(n) = (2^n + n^2 \cdot \log(n)) \cdot (n^3 + n^2 \cdot \log(n))$
(c) $f(n) = (n^5 + 10n^2 + 4n) \cdot (n \cdot \log(n) + 10 \cdot \log(n)) + n^7$
(d) $f(n) = (2^n + 3^n + 100 \cdot n) \cdot (n^n + n! + \log(n))$

(3) Complete the proof of Theorem II.5.1.

(4) Show 2^n $<'$ $n!$ using theorem II.5.5. Hint: Write out the factors of 2^n and $n!$ and give a convincing argument that $2^n / n!$ goes to 0 as n gets arbitrarily large. (Study Example II.5.5.)

(5) Complete the proof of Theorem II.5.7**.**

(6) **(a)** Prove the portion of Theorem II.5.4 for the big O part.
(b) Prove Theorem II.5.4 using the criteria for Θ given by Lemma II.5.1. (This is somewhat more challenging.)