Algorithms

Chapter 3

With Question/Answer Animations

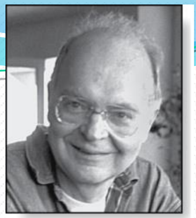
Chapter Summary

- Algorithms
  - Example Algorithms
  - Algorithmic Paradigms
- Growth of Functions
  - Big-O and other Notation
- Complexity of Algorithms

The Growth of Functions

Section 3.2

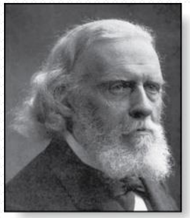
Section Summary



Donald E. Knuth
(Born 1938)

- Big-O Notation
- Big-O Estimates for Important Functions
- Big-Omega and Big-Theta Notation

Edmund Landau
(1877-1938)



Paul Gustav Heinrich Bachmann
(1837-1920)

The Growth of Functions

- In both CS and math:
  - There are times when we care about how fast a function grows.
- In CS, the issue is known as **complexity** (section 3.3). Here are some questions that may arise:
  - How quickly does an algorithm solve a problem as input grows?
  - How does the efficiency of two different algorithms for solving the same problem compare?
  - Is it practical to use a particular algorithm as the input grows?
- In math, growth of functions are studied in
  - number theory (Chapter 4)
  - combinatorics (Chapters 6 and 8)

Big-O Notation

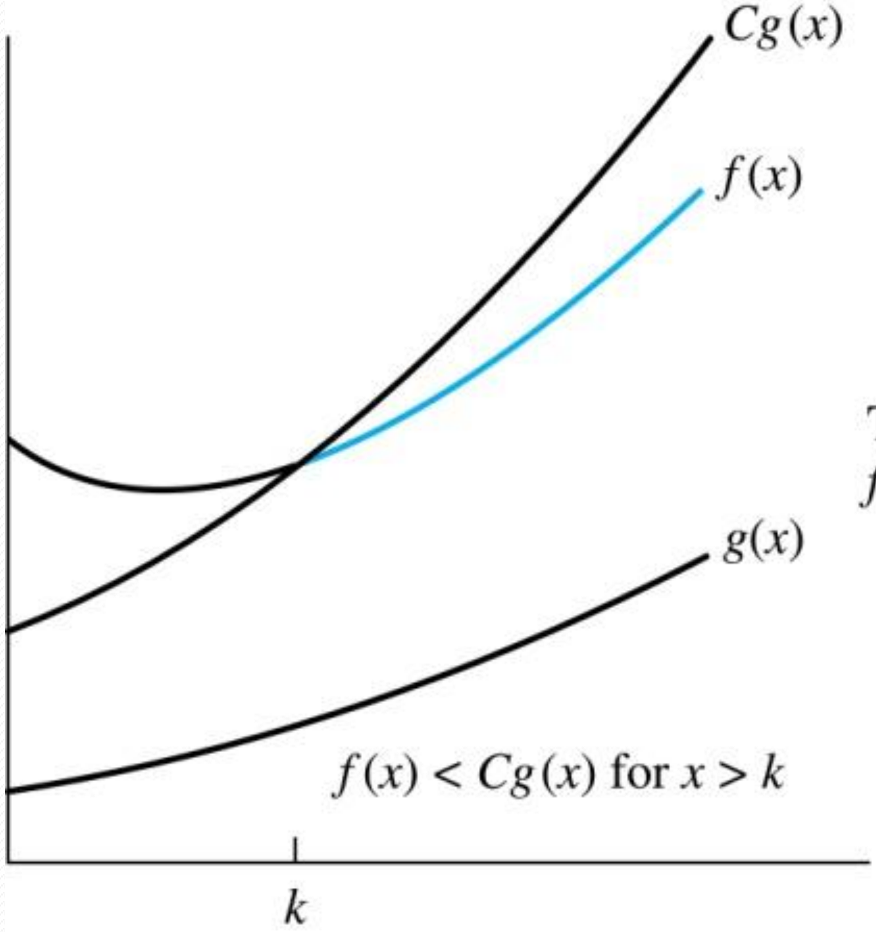
Definition: Let f and g be functions from Z or R to R. $f(x)$ is $\mathbf{O(g(x))}$ if $\exists$constants C and k such that

$$|f(x)| \leq C|g(x)|$$

whenever $x > k$. (illustration on next slide)

- This is read
  - "$f(x)$ is big-O of $g(x)$" or
  - "f is asymptotically dominated by g."
- C and k are *witnesses* to relationship $f(x)$ is $O(g(x))$.
  - Only one pair of witnesses is needed.

Illustration of Big-*O* Notation



$f(x)$ is $O(g(x))$

The part of the graph of $f(x)$ that satisfies $f(x) < Cg(x)$ is shown in color.

Important Points re Big-O Notation

If ∃one pair of witnesses, then ∃infinitely many pairs.

- We can always make k or C larger and still maintain inequality . $|f(x)| \leq C|g(x)|$
- i.e., any pair C' and k' where $C < C'$ and $k < k'$ is also a pair of witnesses since $|f(x)| \leq C|g(x) \leq C'|g(x)|$ whenever $x > k' > k$.

You may see "$f(x) = O(g(x))$" instead of "$f(x)$ is $O(g(x))$."

  - But this is abuse since there is no equality just inequality.

More Important Points re Big-O

- It is ok to write $f(x) \in O(g(x))$, because $O(g(x))$ represents the set of functions that are $O(g(x))$.
- When functions take on positive values only
  - we drop the absolute value sign:

$$f(x) \text{ is } O(g(x)) \text{ if } \exists C, k > 0 \text{ such that } \forall x > k, f(x) \leq Cg(x)$$

Using the Definition of Big-*O* Notation

Example: Show that $f(x) = x^2 + 2x + 1$ is $O(x^2)$.

Solution: Since when $x \geq 1$, $x \leq x^2$ and $1 \leq x^2$

$$0 \leq x^2 + 2x + 1 \leq x^2 + 2x^2 + x^2 = 4x^2$$

- So can take $C = 4$ and $k = 1$ as witnesses

(see graph on next slide)

- Alternatively, when $x \geq 2$, we have $2x \leq x^2$ and $1 \leq x^2$

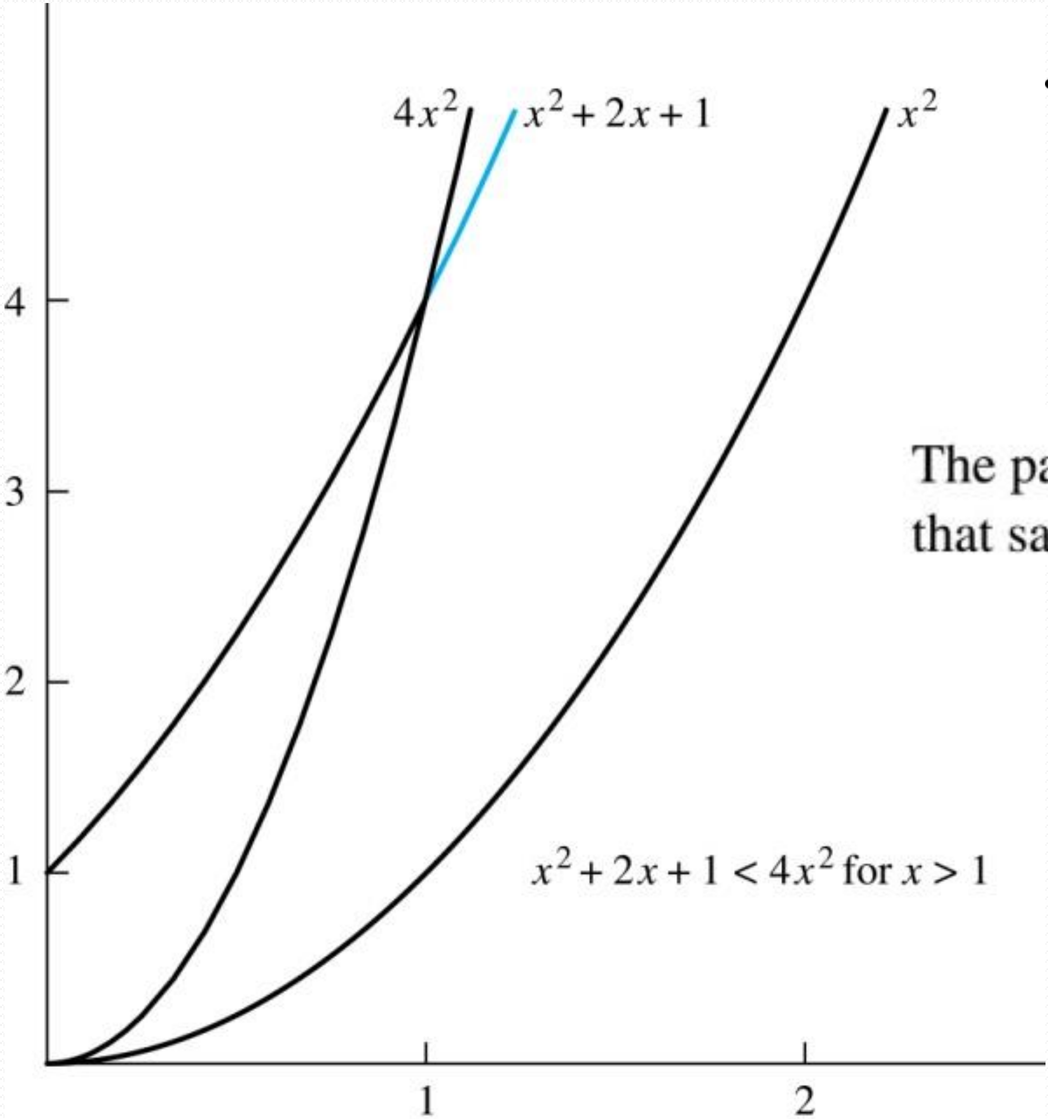
$$0 \leq x^2 + 2x + 1 \leq x^2 + x^2 + x^2 = 3x^2$$

  - So can take $C = 3$ and $k = 2$ as witnesses instead.

Illustration of Big-*O* Notation

$$f(x) = x^2 + 2x + 1$$

is $O(x^2)$



The part of the graph of $f(x) = x^2 + 2x + 1$ that satisfies $f(x) < 4x^2$ is shown in blue.

Big-*O* Notation continued

- Since both $f(x) = x^2 + 2x + 1$ and $g(x) = x^2$ are such that $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$ (why?). We say that the two functions are of the *same order*.
  (More on this later)
- If $f(x)$ is $O(g(x))$ and $\forall x > r$, $h(x) \geq g(x)$, then $f(x)$ is $O(h(x))$.
  [for the witness pair, choose the same C and let $k' = \max(k, r)$]
- For many applications, the goal is to select the function $g(x)$ in $O(g(x))$ as small as possible (up to multiplication by a constant, of course).

Using the Definition of Big-O Notation

Example: Show that $7x^2$ is $O(x^3)$.

Solution: When $x > 7$, $7x^2 < x^3$. Take $C = 1$ and $k = 7$ as witnesses to establish that $7x^2$ is $O(x^3)$.

(Would $C = 7$ and $k = 1$ work?)

Example: Show that n^2 is not $O(n)$.

Solution: Suppose $\exists C, k$ for which $n^2 \leq Cn$, whenever $n > k$. Then (by dividing both sides of $n^2 \leq Cn$) by n, then $n \leq C$ must hold for all $n > k$. A contradiction!

Big-*O* Estimates for Polynomials

Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_o$

where $a_0, a_1, \ldots, a_n$ are real numbers with $a_n \neq 0$.

Then $f(x)$ is $O(x^n)$.

Proof: $|f(x)| = |a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x^1 + a_0|$

Use triangle inequality, exercise, Section 1.8.

$\leq |a_n| x^n + |a_{n-1}| x^{n-1} + \cdots + |a_1| x^1 + |a_0|$

$= x^n (|a_n| + |a_{n-1}| / x + \cdots + |a_1| / x^{n-1} + |a_0| / x^n)$

Assuming $x > 1$

$\leq x^n (|a_n| + |a_{n-1}| + \cdots + |a_1| + |a_0|)$

Take $C = |a_n| + |a_{n-1}| + \cdots + |a_0|$ and $k = 1$. QED

- Leading term $a_n x^n$ of polynomial *dominates* its growth.

More Big-*O* Estimates

Example: Use big-*O* notation to estimate the sum of the first *n* positive integers.

Solution: $1 + 2 + \cdots + n \ \leq \ n + n + \cdots n \ = \ n^2$

Hence $1 + 2 + \ldots + n$ is $O(n^2)$ taking $C = 1$ and $k = 1$.

More Big-O Estimates (cont)

Example: Use big-O notation to estimate n! and log n!

Solution: $f(n) = n! = 1 \times 2 \times \cdots \times n$.

$$*n! = 1 \times 2 \times \cdots \times n \leq n \times n \times \cdots \times n = n^n$$

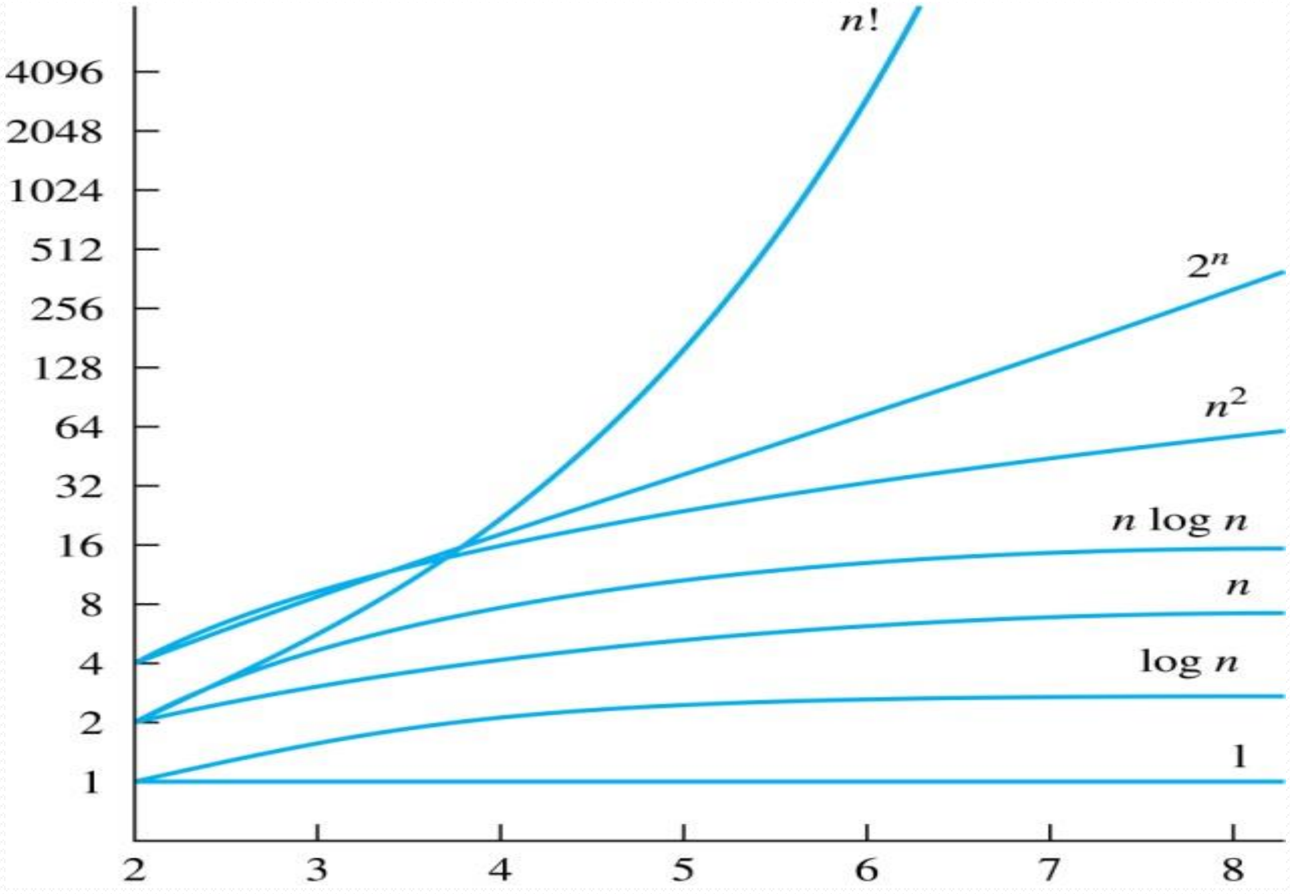
Hence $n!$ is $O(n^n)$ taking $C = 1$ and $k = 1$.

Now apply log to * and use a property of logarithms

$$\log(n!) \leq n \cdot \log(n)$$

Hence, $\log(n!)$ is $O(n \cdot \log(n))$ taking $C = 1$ and $k = 1$.

Display of Growth of Functions



Note the difference in behavior of functions as *n* gets larger

Who dominates who among logarithms, powers, and exponents?

- If $d > c > 1$, then n^c is $O(n^d)$, but n^d is not $O(n^c)$.
- If $c > b > 1$, then b^n is $O(c^n)$, but c^n is not $O(b^n)$.

  [exponentials and powers strictly dominate within their classes as expected]
- If $b > 1$ and c and d are positive, then

  $(\log_b n)^c$ is $O(n^d)$, but n^d is not $O((\log_b n)^c)$

  [any power strictly dominates a log power]
- If $b > 1$ & $d > 0$, then n^d is $O(b^n)$, but b^n is not $O(n^d)$.

  [any exponential strictly dominates a power]

Combinations of Functions

- If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then
$$(f_1 + f_2)(x) \text{ is } O(\max(|g_1(x)|,|g_2(x)|)).$$
(See next slide for proof)
- If $f_1(x)$ and $f_2(x)$ are both $O(g(x))$ then
$$(f_1 + f_2)(x) \text{ is } O(g(x)).$$
(See text for argument)
- If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then
$$(f_1 f_2)(x) \text{ is } O(g_1(x)g_2(x)).$$
(See text for argument)

Combinations of Functions

If $f_1(x)$ is $O(g_1(x))$ and $f_2(x)$ is $O(g_2(x))$ then

$$(f_1 + f_2)(x) \text{ is } O(\max(|g_1(x)|,|g_2(x)|)).$$

Proof:

By the definition of big-O notation, $\exists C_1, C_2, k_1, k_2$ such that
$|f_1(x)| \le C_1|g_1(x)|$ when $x > k_1$ and $|f_2(x)| \le C_2|g_2(x)|$ when $x > k_2$.

$$\begin{aligned}
|(f_1 + f_2)(x)| &= |f_1(x) + f_2(x)| \\
&\le |f_1(x)| + |f_2(x)| && \text{by the triangle inequality } |a + b| \le |a| + |b| \\
&\le C_1|g_1(x)| + C_2|g_2(x)| \\
&\le C_1 g(x) + C_2 g(x) && \text{where } g(x) = \max(|g_1(x)|,|g_2(x)|) \\
&= (C_1 + C_2)\, g(x) \\
&= C\, g(x) && \text{where } C = C_1 + C_2
\end{aligned}$$

Therefore $|(f_1 + f_2)(x)| \le C\, g(x)$ whenever $x > k$, where $k = \max(k_1,k_2)$

Ordering Functions by Order of Growth

Put in order so that each function is big-O of next function:

- $f_1(n) = (1.5)^n$
- $f_2(n) = 8n^3+17n^2+111$
- $f_3(n) = (\log n)^2$
- $f_4(n) = 2^n$
- $f_5(n) = \log(\log n)$
- $f_6(n) = n^2 (\log n)^3$
- $f_7(n) = 2^n (n^2+1)$
- $f_8(n) = n^3 + n(\log n)^2$
- $f_9(n) = 10000$
- $f_{10}(n) = n!$
- Start by finding the dominant term in the 2 functions that have multiple terms.
- Use hierarchy--constant, log of log, powers of log, powers, exponential, factorial (n^n)--to put into categories.
- Follow the usually clear hierarchy within each category.

Ordering Functions by Order Solution

$f_9(n) = 10000$ (constant, does not increase with n)

$f_5(n) = \log(\log n)$ (grows slowest of all the others)

$f_3(n) = (\log n)^2$ (grows next slowest)

$f_6(n) = n^2 (\log n)^3$ (next largest, $(\log n)^3$ factor smaller than any power of n)

$f_2(n) = 8n^3 + 17n^2 + 111$ (tied with the one below)

$f_8(n) = n^3 + n(\log n)^2$

$f_1(n) = (1.5)^n$ (next largest, an exponential function)

$f_4(n) = 2^n$ (grows faster than one above since $2 > 1.5$)

$f_7(n) = 2^n (n^2 + 1)$ (grows faster than above because of the $n^2 + 1$ factor)

$f_{10}(n) = n!$ ($n!$ grows faster than c^n for every c)

Big-Omega Notation

Definition: Let *f* and *g* be functions from Z or R to R.

$f(x)$ is $\Omega(g(x))$ if $\exists C$ and k such that

$$|f(x)| \geq C|g(x)| \quad \text{when } x > k.$$

Ω is the upper case version of the lower case Greek letter ω.

"*f*(*x*) is big-Omega of *g*(*x*)" or "*f* asymptotically dominates *g*."

- Big-*O* gives an upper bound on the growth of a function, while Big-Omega gives a lower bound.
- Big-Omega tells us that a function grows at least as fast as another.

Big-Omega Notation

Example: Show that $f(x) = 8x^3 + 5x^2 + 7$ is $\Omega(g(x))$ where $g(x) = x^3$.

Solution: $f(x) = 8x^3 + 5x^2 + 7 \geq 8x^3 \ \forall x \in R.$

$\therefore$ $f(x)$ is $\Omega(g(x))$ (Take $C = 8$ and $k = 1$)

- Is it also the case that $g(x) = x^3$ is $O(8x^3 + 5x^2 + 7)$?
- What can you take for C and k?
- Can we generalize this observation?

- $f(x)$ is $\Omega(g(x))$ if and only if $g(x)$ is $O(f(x))$.
  - This follows from the definitions. (See text for details.)
  - If pair for LHS is C, k, we can take as pair for RHS 1/C, k.

Big-Theta Notation

Θ is the upper case version of the lower case Greek letter θ.

Definition: Let *f* and *g* be functions from Z or R to R.

$f(x)$ is $\Theta(g(x))$ if $f(x)$ is $O(g(x))$ and $f(x)$ is $\Omega(g(x))$.

- We say that
  - "*f*(*x*) is big-Theta of *g*(*x*)"
  - "*f*(*x*) is of *order* *g*(*x*)"
  - "*f*(*x*) and *g*(*x*) are of the *same order*."
- $f(x)$ is $\Theta(g(x))$ if and only if $\exists$constants C_1, C_2 and k such that $C_1 g(x) < f(x) < C_2 g(x) \;\; \forall\, x > k$.

  (This follows from the definitions of big-*O* and big-Omega.)

Big Theta Notation example 1

Example: Show that sum of first n positive integers is $\Theta(n^2)$.

Solution: Let $f(n) = 1 + 2 + \cdots + n$.

- We have already shown that $f(n)$ is $O(n^2)$.
- To show that $f(n)$ is $\Omega(n^2)$, we need a positive constant C such that $f(n) > Cn^2$ for sufficiently large n. Summing only the terms $\geq n/2$ we obtain the inequality. To ease calculations, we assume n even. We leave n odd case as exercise.

$$1 + 2 + \cdots + n \geq n/2 + (n/2 + 1) + \cdots + n$$
$$\geq n/2 + n/2 + \cdots + n/2$$
$$= (n/2 + 1)(n/2) \geq n^2/4$$

- Taking $C = 1/4$, $f(n) > Cn^2 \; \forall n \in Z^+$. Hence, $f(n)$ is $\Omega(n^2)$.
- $\therefore f(n)$ is $\Theta(n^2)$.

Big-Theta Notation example 2

Example: Show that $f(x) = 3x^2 + 8x \log x$ is $\Theta(x^2)$.

Solution:

- $3x^2 + 8x \log x \leq 11x^2$ for $x > 1$,
  since $0 \leq 8x \log x \leq 8x^2$.
  - Hence, $3x^2 + 8x \log x$ is $O(x^2)$.

  (Why? What pair C, k have we shown to work?)
- $3x^2 + 8x \log x$ is clearly $\Omega(x^2)$.

(Why? What pair C, k works?)

- Hence, $3x^2 + 8x \log x$ is $\Theta(x^2)$.

Miscellaneous Θ facts/confusion

- If $f(x)$ is $\Theta(g(x))$ then $g(x)$ is $\Theta(f(x))$ as well.
- Also note that $f(x)$ is $\Theta(g(x))$ if and only if

  $f(x)$ is $O(g(x))$ and $g(x)$ is $O(f(x))$

  - partially accounting for why you see big-Omega infrequently.
- Writers are often careless and use big-*O* when they really mean big-Teta.

Big-Theta Estimates for Polynomials

Theorem: Let $f(x) = a_n x^n + a_{n-1} x^{n-1} + \cdots + a_1 x + a_o$ where $a_0, a_1, \ldots, a_n$ are real numbers with $a_n \neq 0$.

Then $f(x)$ is $\Theta(x^n)$ (or of order x^n).

(The proof is an exercise.)

Examples:

$f(x) = 8x^5 + 5x^2 + 10$ is $\Theta(x^5)$.

$f(x) = 8x^{199} + 7x^{100} + x^{99} + 5x^2 + 25$ is $\Theta(x^{199})$.

Classifying Functions by their Order

$f_9(n) = 10000$ is	$\Theta(1)$
$f_5(n) = \log(\log n)$ is	$\Theta(\log(\log n))$
$f_3(n) =$ is	$\Theta((\log n)^2)$
$f_6(n) = n^2 (\log n)^3$ is	$\Theta(n^2 (\log n)^3)$
$f_2(n) = 8n^3 + 17n^2 + 111 n^2$ is	$\Theta(n^3)$
$f_8(n) = n^3 + n(\log n)^2$ is	$\Theta(n^3)$
$f_1(n) = (1.5)^n$ is	$\Theta((1.5)^n)$
$f_4(n) = 2^n$ is	$\Theta(2^n)$
$f_7(n) = 2^n (n^2 + 1)$ is	$\Theta(n^2 2^n)$
$f_{10}(n) = n!$ is	$\Theta(n^n)$