Announcements

- Please sit in the front half
- Reading Assignments:
  Chapter 4 (Textbook: CLRS)
- Please turn in the survey form

Asymptotic Notation

- **Θ, O, Ω, o, ω**
- **Used to describe the running times of algorithms**
- **Instead of exact running time, say $\Theta(n^2)$**
- **Defined for functions whose domain is the set of natural numbers**
- **Determine *sets* of functions, in practice used to compare two functions**

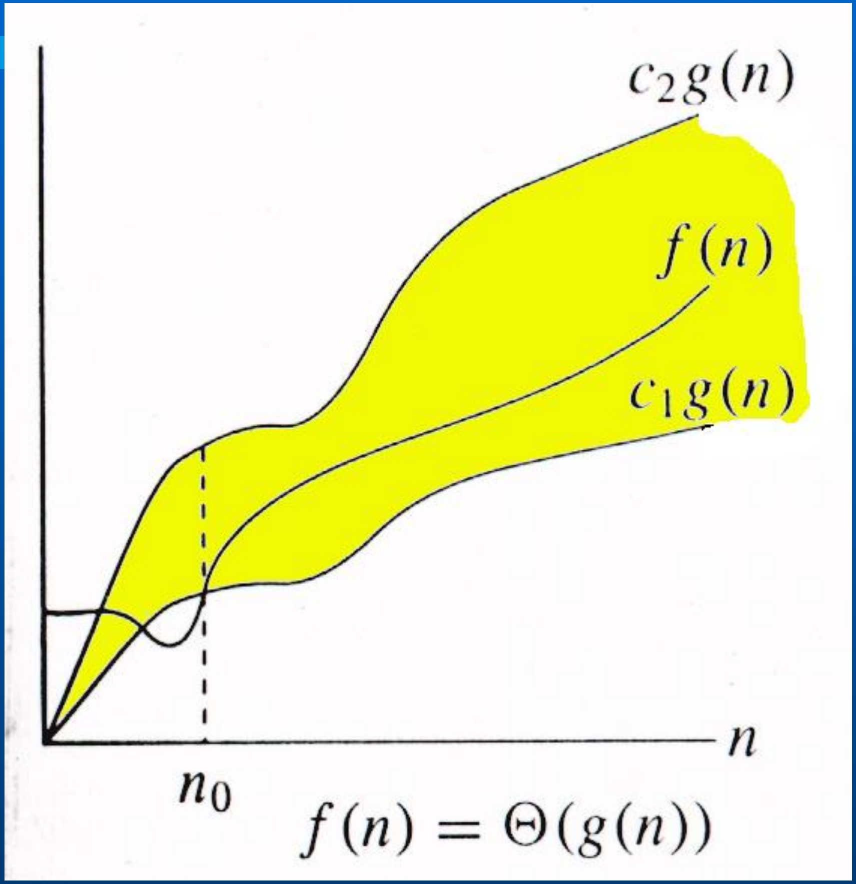
Θ-notation

For a given function $g(n)$, we denote by $\Theta(g(n))$ the set of functions

$\Theta(g(n)) = \{f(n)$: there exist positive constants c_1, c_2 and n_0 such that

$0 \le c_1 g(n) \le f(n) \le c_2 g(n)$,

for all $n \ge n_0\}$



We say $g(n)$ is an *asymptotically tight bound* for $f(n)$

Example

- $10n^2 - 3n = \Theta(n^2)$
- **What constants for n_0, c_1, and c_2 will work?**
- **Make c_1 a little smaller than leading coefficient, and c_2 a little bigger**
- ***To compare orders of growth, look at leading term***

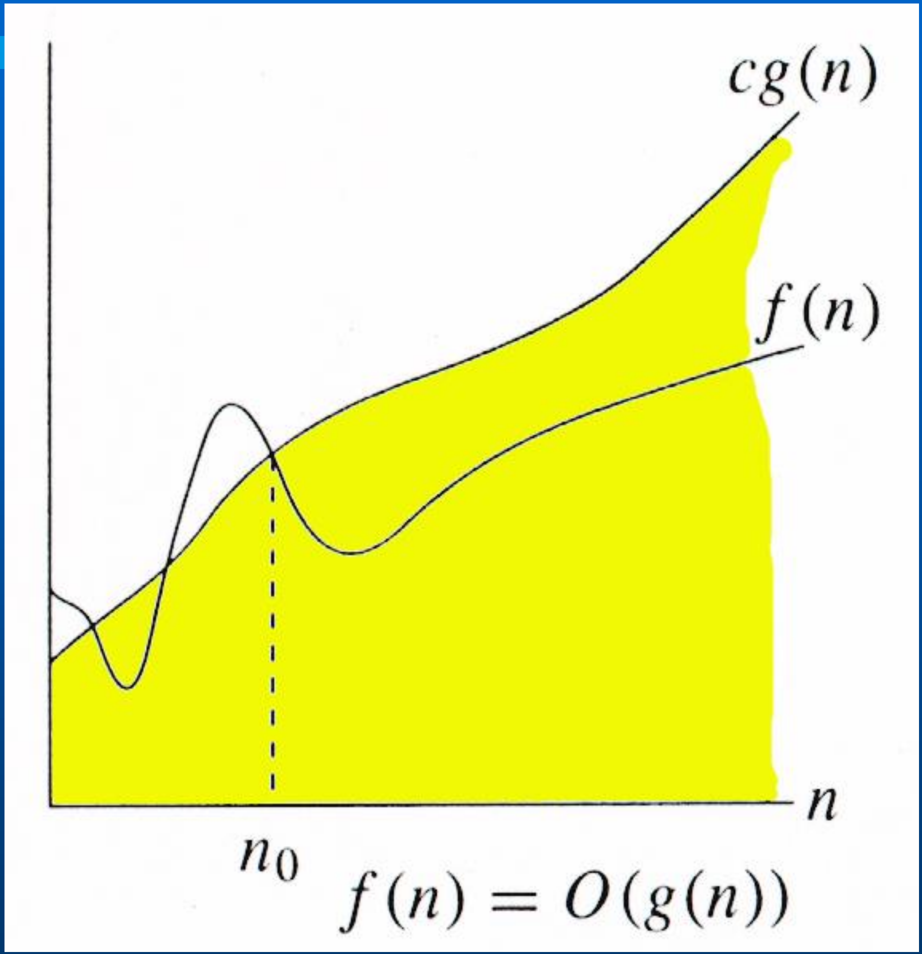
O-notation

For a given function $g(n)$, we denote by $O(g(n))$ the set of functions

$O(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that

$0 \le f(n) \le cg(n)$

for all $n \ge n_0$ $\}$



We say $g(n)$ is an *asymptotic upper bound* for $f(n)$

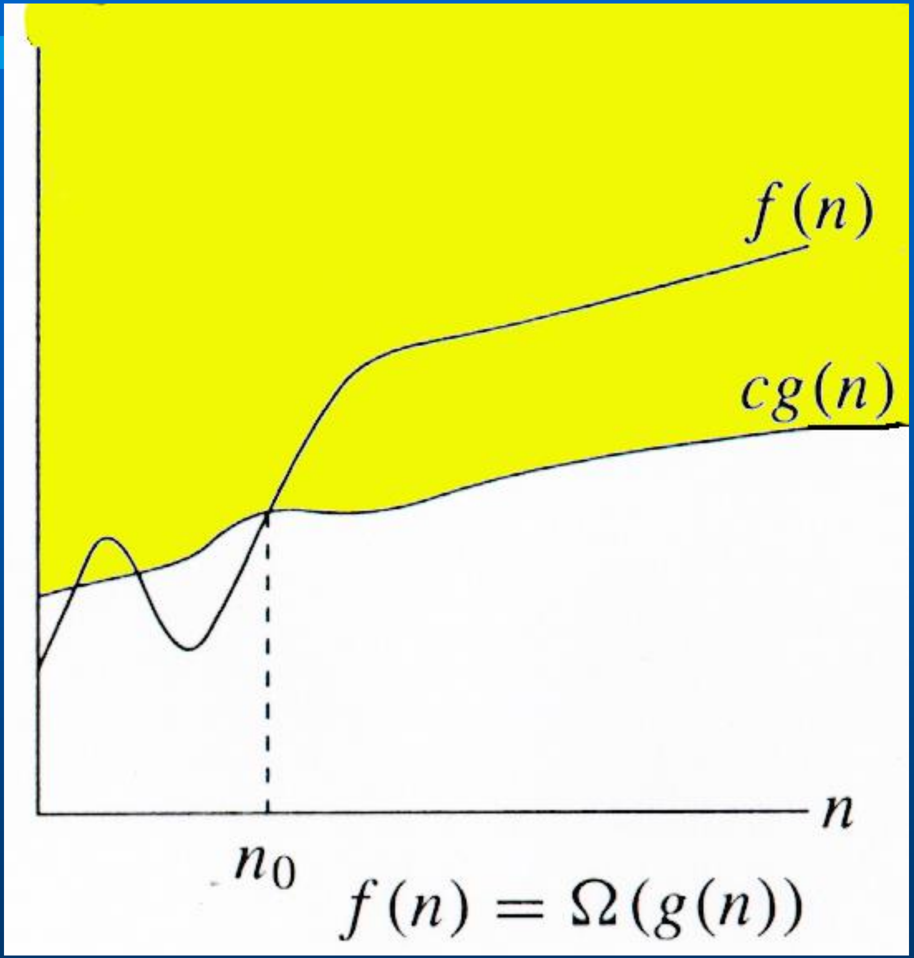
Ω-notation

For a given function $g(n)$, we denote by $\Omega(g(n))$ the set of functions

$\Omega(g(n)) = \{f(n)$: there exist positive constants c and n_0 such that

$0 \leq cg(n) \leq f(n)$

for all $n \geq n_0\}$



We say $g(n)$ is an *asymptotic lower bound* for $f(n)$

Relations Between Θ, Ω, O

- For any two functions $g(n)$ and $f(n)$, $f(n) = \Theta(g(n))$ if and only if $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

- I.e., $\Theta(g(n)) = O(g(n)) \cap \Omega(g(n))$

Running Times

- "The running time is $O(f(n))$" $\Rightarrow$ Worst case is $O(f(n))$
- "Running time is $\Omega(f(n))$" $\Rightarrow$ Best case is $\Omega(f(n))$
- Can still say "Worst case running time is $\Omega(f(n))$"
  - Means worst case running time is given by some unkown function $g(n) \in \Omega(f(n))$

Example

- ***Insertion sort*** **takes $\Theta(n^2)$ worst case time, so sorting (as a *problem*) is $O(n^2)$**
- **Any sort algorithm must look at each item, so sorting is $\Omega(n)$**
- **In fact, using (e.g.) merge sort, sorting is $\Theta(n \lg n)$ in the worst case**

Asymptotic Notation in Equations

- Used to replace expressions containing lower-order terms
- For example,

  $4n^3 + 3n^2 + 2n + 1 = 4n^3 + 3n^2 + \Theta(n)$

  $= 4n^3 + \Theta(n^2) = \Theta(n^3)$

- In equations, $\Theta(f(n))$ always stands for an ***anonymous function*** $g(n) \in \Theta(f(n))$
  - In the example above, $\Theta(n^2)$ stands for $3n^2 + 2n + 1$

o-notation

For a given function $g(n)$, we denote by $o(g(n))$ the set of functions

$o(g(n)) = \{f(n)$: for any positive constant $c > 0$, there exists a constant $n_0 > 0$ such that $0 \leq f(n) < cg(n)$ for all $n \geq n_0\}$

$f(n)$ becomes insignificant relative to $g(n)$ as n approaches infinity: $\lim_{n\to\infty} [f(n) / g(n)] = 0$

We say $g(n)$ is an *upper bound* for $f(n)$ that is not asymptotically tight.

ω-notation

For a given function $g(n)$, we denote by $\omega(g(n))$ the set of functions

$\omega(g(n)) = \{f(n)$: for any positive constant $c > 0$, there exists a constant $n_0 > 0$ such that $0 \leq cg(n) < f(n)$ for all $n \geq n_0$ $\}$

$f(n)$ becomes arbitrarily large relative to $g(n)$ as n approaches infinity: $\lim_{n \to \infty} [f(n) / g(n)] = \infty$

We say $g(n)$ is a *lower bound* for $f(n)$ that is not asymptotically tight.

Limits

- $\lim_{n \to \infty} [f(n) / g(n)] = 0 \Rightarrow f(n) \in o(g(n))$
- $\lim_{n \to \infty} [f(n) / g(n)] < \infty \Rightarrow f(n) \in O(g(n))$
- $0 < \lim_{n \to \infty} [f(n) / g(n)] < \infty \Rightarrow f(n) \in \Theta(g(n))$
- $0 < \lim_{n \to \infty} [f(n) / g(n)] \Rightarrow f(n) \in \Omega(g(n))$
- $\lim_{n \to \infty} [f(n) / g(n)] = \infty \Rightarrow f(n) \in \omega(g(n))$
- $\lim_{n \to \infty} [f(n) / g(n)]$ undefined $\Rightarrow$ can't say

Comparison of Functions

$$f \leftrightarrow g \;\approx\; a \leftrightarrow b$$

$$f(n) = O(g(n)) \;\approx\; a \le b$$

$$f(n) = \Omega(g(n)) \;\approx\; a \ge b$$

$$f(n) = \Theta(g(n)) \;\approx\; a = b$$

$$f(n) = o(g(n)) \;\approx\; a < b$$

$$f(n) = \omega(g(n)) \;\approx\; a > b$$

Properties

- **Transitivity**

$$f(n) = \Theta(g(n)) \ \& \ g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$$
$$f(n) = O(g(n)) \ \& \ g(n) = O(h(n)) \Rightarrow f(n) = O(h(n))$$
$$f(n) = \Omega(g(n)) \ \& \ g(n) = \Omega(h(n)) \Rightarrow f(n) = \Omega(h(n))$$
$$f(n) = o\,(g(n)) \ \& \ g(n) = o\,(h(n)) \Rightarrow f(n) = o\,(h(n))$$
$$f(n) = \omega(g(n)) \ \& \ g(n) = \omega(h(n)) \Rightarrow f(n) = \omega(h(n))$$

- **Symmetry**

$f(n) = \Theta(g(n))$ **if and only if** $g(n) = \Theta(f(n))$

- **Transpose Symmetry**

$f(n) = O(g(n))$ **if and only if** $g(n) = \Omega(f(n))$

$f(n) = o(g(n))$ **if and only if** $g(n) = \omega((f(n))$

Useful Facts

- For $a \geq 0$, $b > 0$, $\lim_{n\to\infty} (\lg^a n / n^b) = 0$, so $\lg^a n = o(n^b)$, and $n^b = \omega(\lg^a n)$
  - Prove using L'Hopital's rule repeatedly
- $\lg(n!) = \Theta(n \lg n)$

Examples

	A	B
•	$5n^2 + 100n$	$3n^2 + 2$
•	$\log_3(n^2)$	$\log_2(n^3)$
•	$n^{\lg 4}$	$3^{\lg n}$
•	$\lg^2 n$	$n^{1/2}$

Examples

	A	B	
•	$5n^2 + 100n$	$3n^2 + 2$	$A \in \Theta(B)$
	$A \in \Theta(n^2),\ n^2 \in \Theta(B) \Rightarrow A \in \Theta(B)$		
•	$\log_3(n^2)$	$\log_2(n^3)$	
•	$n^{\lg 4}$	$3^{\lg n}$	
•	$\lg^2 n$	$n^{1/2}$	

Examples

	A	B	
•	$5n^2 + 100n$	$3n^2 + 2$	$A \in \Theta(B)$

$A \in \Theta(n^2),\ n^2 \in \Theta(B) \Rightarrow A \in \Theta(B)$

	A	B	
•	$\log_3(n^2)$	$\log_2(n^3)$	$A \in \Theta(B)$

$\log_b a = \log_c a / \log_c b$; $A = 2\lg n / \lg 3$, $B = 3\lg n$, $A/B = 2/(3\lg 3)$

	A	B
•	$n^{\lg 4}$	$3^{\lg n}$
•	$\lg^2 n$	$n^{1/2}$

Examples

A B

- $5n^2 + 100n$ $3n^2 + 2$ $A \in \Theta(B)$

  $A \in \Theta(n^2),\ n^2 \in \Theta(B) \Rightarrow A \in \Theta(B)$

- $\log_3(n^2)$ $\log_2(n^3)$ $A \in \Theta(B)$

  $\log_b a = \log_c a \,/\, \log_c b$; $A = 2\lg n \,/\, \lg 3$, $B = 3\lg n$, $A/B = 2/(3\lg 3)$

- $n^{\lg 4}$ $3^{\lg n}$ $A \in \omega(B)$

  $a^{\log b} = b^{\log a}$; $B = 3^{\lg n} = n^{\lg 3}$; $A/B = n^{\lg(4/3)} \to \infty$ as $n \to \infty$

- $\lg^2 n$ $n^{1/2}$

Examples

	A	B	
•	$5n^2 + 100n$	$3n^2 + 2$	$A \in \Theta(B)$
	$A \in \Theta(n^2),\ n^2 \in \Theta(B) \Rightarrow A \in \Theta(B)$		
•	$\log_3(n^2)$	$\log_2(n^3)$	$A \in \Theta(B)$
	$\log_b a = \log_c a / \log_c b$; $A = 2\lg n / \lg 3$, $B = 3\lg n$, $A/B = 2/(3\lg 3)$		
•	$n^{\lg 4}$	$3^{\lg n}$	$A \in \omega(B)$
	$a^{\log b} = b^{\log a}$; $B = 3^{\lg n} = n^{\lg 3}$; $A/B = n^{\lg(4/3)} \to \infty$ as $n \to \infty$		
•	$\lg^2 n$	$n^{1/2}$	$A \in o(B)$
	$\lim_{n\to\infty} (\lg^a n / n^b) = 0$ (here $a = 2$ and $b = 1/2$) $\Rightarrow A \in o(B)$		

Review on Summation

- **Why do we need summation formulas?**

  We need them for computing the running times of some algorithms. (CLRS/Chapter 3)

- **Example: Maximum Subvector**

  Given an array $a[1\ldots n]$ of numeric values (can be positive, zero and negative) determine the subvector $a[i\ldots j]$ $(1 \leq i \leq j \leq n)$ whose sum of elements is maximum over all subvectors.

1	-2	2	2

Reviews on Summation

```
MaxSubvector(a, n)
    maxsum ← 0;
    for i ← 1 to n
        for j = i to n
            sum ← 0;
            for k ← i to j
                sum += a[k]
            maxsum ← max(sum, maxsum);
    return maxsum;
```

- $T(n) = \sum_{i=1}^{n} \sum_{j=i}^{n} \sum_{k=i}^{j} 1$
- **NOTE: This is not a simplified solution. What *is* the final answer?**

Reviews on Summation

- **Constant Series: For $n \geq 0$,**

$$\sum_{i=a}^{b} 1 = b - a + 1$$

- **Linear Series: For $n \geq 0$,**

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

Reviews on Summation

- **Quadratic Series: For $n \geq 0$,**

$$\sum_{i=1}^{n} i^2 = 1^2 + 2^2 + \ldots + n^2 = (2n^3 + 3n^2 + n) / 6$$

- **Linear-Geometric Series: For $n \geq 0$,**

$$\sum_{i=1}^{n} ic^i = c + 2c^2 + \ldots + nc^n = [(n-1)c^{(n+1)} - nc^n + c] / (c-1)^2$$