Three time complexity functions

- The **worse case** time complexity of the algorithm is the function defined by the maximum number of operations performed, taken across all instances of size n.
- The **best case** time complexity of the algorithm is the function defined by the minimum number of operations performed, taken across all instances of size n.
- The **average-case** time complexity of the algorithm is the function defined by an average number of operations performed, taken across all instances of size n.

Which complexity is most important?

1. The worst-case time complexity is an upper bound on the running time for any input.
2. For some algorithms, the worst case occurs fairly often. Binary search is an example.
3. Usually for a good algorithm, then average case is often roughly as bad as the worse case. Insertion sort is an example.

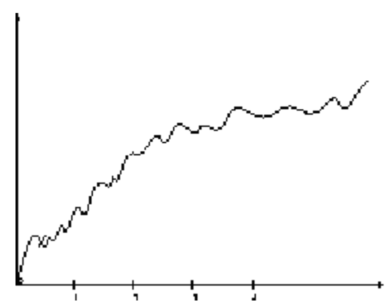
Exact analysis is hard and not necessary (in most cases)

The exact complexity of insertion sort is:

$$T(n) = (\frac{c_5}{2} + \frac{c_6}{2} + \frac{c_7}{2})n^2 + (c_1 + c_2 + c_4 + \frac{c_5}{2} - \frac{c_6}{2} - \frac{c_7}{2} + c_8)n - (c_2 + c_4 + c_5 + c_8)$$

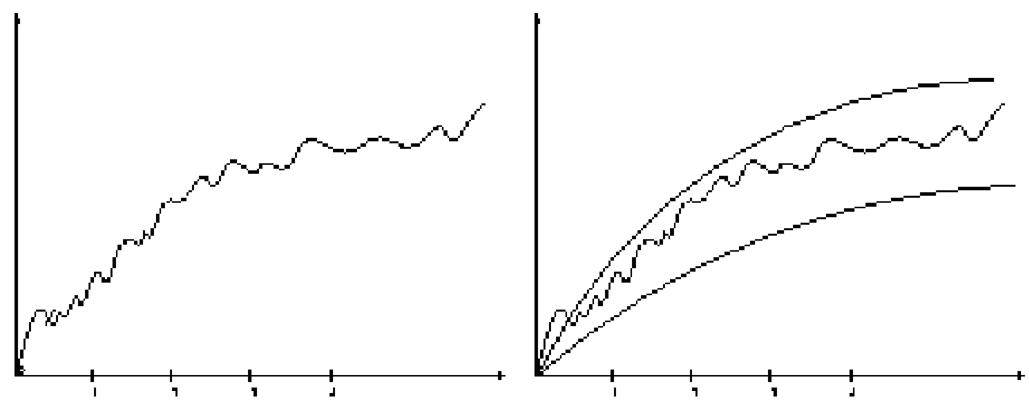
we prefer to write it as $an^2 + bn + c$

Another example of exact function $n^3 + 100n + 2 + 10\lg n + 0.1 * 2^n$ may look like this:



What should we do? Simplify the analysis

- Ignore the constants:
  - We can erase the difference between $T_1(n) = 2n$ and $T_2(n) = 8n$ by using different machines.
- Only look at "dominant" terms:
  - $n^3 + 100n + 2$ is "dominated" by n^3 as $n \to \infty$



Asymptotic Notation: order of growth

- $f(n) = O(g(n))$ means $C \times g(n)$ is an asymptotic **upper bound** on $f(n)$.

- $f(n) = \Omega(g(n))$ means $C \times g(n)$ is an asymptotic **lower bound** on $f(n)$.

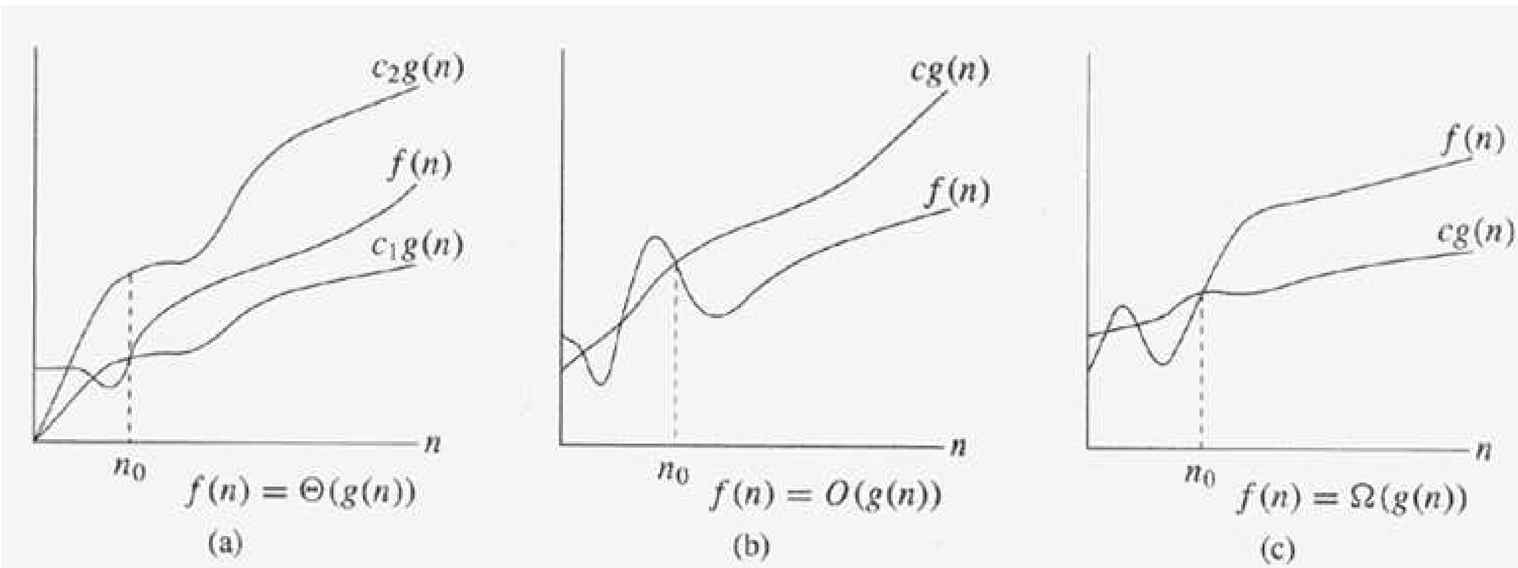
- $f(n) = \Theta(g(n))$ means $C_1 \times g(n)$ is an asymptotic **upper bound** on $f(n)$ and $C_2 \times g(n)$ is an asymptotic **lower bound** on $f(n)$.

O, Ω and Θ notations

Definition:

- $f(n) = O(g(n))$ if $\exists$ positive constants n_0 and c such that $0 \leq f(n) \leq cg(n)$ for all $n \geq n_0$.
- $f(n) = \Omega(g(n))$ if $\exists$ positive constants n_0 and c such that $f(n) \geq cg(n) \geq 0$ for all $n \geq n_0$.
- $f(n) = \Theta(g(n))$ if $\exists$ positive constants c_1, c_2, and n_0 such that $0 \leq c_1 g(n) \leq f(n) \leq c_2 g(n)$ for all $n \geq n_0$.

O, Ω and Θ notations, Cont'd



Examples

Example 1: $f(n) = 3n^2, g(n) = 10n^2 + 5n$

Claim: $f(n) = O(g(n))$

Proof:

$c = 1; n_0 = 1$ $3n^2 \leq 10n^2 + 5n$ for all $n >= 1$

Examples (Cont'd)

Example 2: $f(n) = 10n^2 + 5n, g(n) = 3n^2$

Claim: $f(n) = O(g(n))$

Proof: $c = 4; 10n^2 + 5n \leq 12n^2$, for all $n \geq 10$

Example 3: $10n^2 + 4n + 2 = O(n^2)$

Proof: $10n^2 + 4n + 2 \leq 11n^2$ for all $n \geq 5$.

Example (Cont'd)

Example 4: $f(n) = a_d n^d + a_{d-1} n^{d-1} + ... + a_1 n + a_0$,

Claim: $f(n) = O(n^d)$

Proof:

$$
\begin{aligned}
f(n) &= a_d n^d + a_{d-1} n^{d-1} + ... + a_1 n + a_0 \\
&\leq |a_d| n^d + |a_{d-1}| n^{d-1} + ... + |a_1| n + |a_0| \\
&= \sum_{i=0}^{d} |a_i| n^i \\
&= n^d \sum_{i=0}^{d} |a_i| n^{i-d} \\
&\leq n^d \sum_{i=0}^{d} |a_i| \text{ for } n \geq 1, \text{ because } n^{i-d} \leq 1 \text{ when } n \geq 1.
\end{aligned}
$$

Alternative Definitions

Definition:

$$f(n) = O(g(n)) \iff \lim_{n\to\infty} \frac{f(n)}{g(n)} = c;$$

where c is a constant $c \geq 0$.

Definition:

$$f(n) = \Omega(g(n)) \iff \lim_{n\to\infty} \frac{g(n)}{f(n)} = c;$$

where c is a constant $c \geq 0$.

Theorem: $f(n) = O(g(n)) \iff g(n) = \Omega(f(n))$.

Examples

Claim: $12n^2 + 5n = \Theta(100n^2)$

Proof:

$$lim_{n\to\infty}\frac{12n^2+5n}{100n^2} = lim_{n\to\infty}\frac{12}{100} + lim_{n\to\infty}\frac{5}{100n} = \frac{12}{100}. \quad (1)$$

Examples (Cont'd)

Claim: $10n^2 + 5n = \Theta(n^3)$?

Proof:

$$lim_{n\to\infty}\frac{10n^2 + 5n}{n^3} = lim_{n\to\infty}\frac{10}{n} + lim_{n\to\infty}\frac{5}{n^2} = 0 + 0 = 0$$

But:

$$lim_{n\to\infty}\frac{n^3}{10n^2 + 5n} = \frac{1}{lim_{n\to\infty}\frac{10n^2+5n}{n^3}} = \infty$$

Hence, $10n^2 + 5n$ is only $O(n^3)$.

Examples (Cont'd)

Compare: $2^n, n^2$ and lg n.

Examples, Cont'd

$2n^2 - 10n + 5 = O(n^2)$, (Yes)
$2n^2 - 10n + 5 = O(n^3)$, (Yes)
$2n^2 - 10n + 5 = O(n)$, (No)

$2n^2 - 10n + 5 = \Omega(n^2)$, (Yes)
$2n^2 - 10n + 5 = \Omega(n^3)$, (No)
$2n^2 - 10n + 5 = \Omega(n)$, (Yes)

$2n^2 - 10n + 5 = \Theta(n^2)$, (Yes)
$2n^2 - 10n + 5 = \Theta(n^3)$, (No)
$2n^2 - 10n + 5 = \Theta(n)$, (No)