Asymptotic notation

1. Pseudo-code – describe algorithms
2. Asymptotic notation – discuss efficiency
3. Design techniques – design algorithms

- Describe *growth* of functions.
- Focus on what's important by abstracting away low-order terms and constant factors.

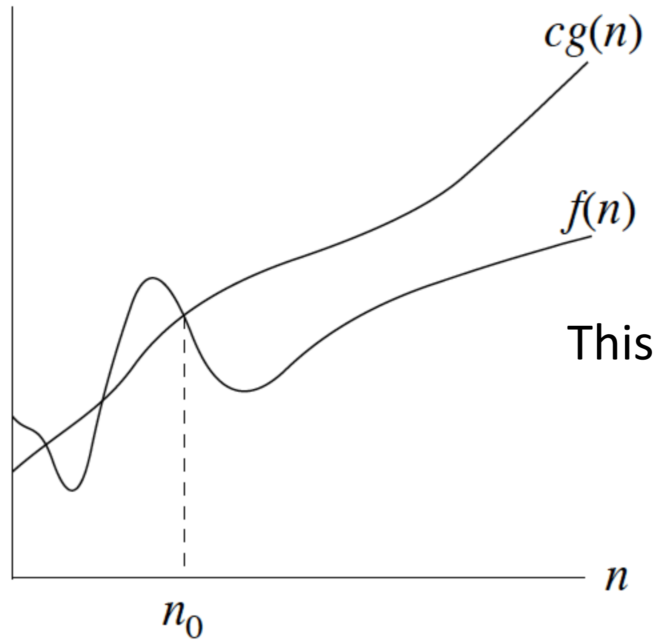
How we indicate running times of algorithms.

A way to compare "sizes" of functions:

$$\begin{array}{ccc} O & \approx & \leq \\ \Omega & \approx & \geq \\ \Theta & \approx & = \\ o & \approx & < \\ \omega & \approx & > \end{array}$$

Asymptotic notation

$$O(g(n)) = \{f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le f(n) \le cg(n) \text{ for all } n \ge n_0\}.$$



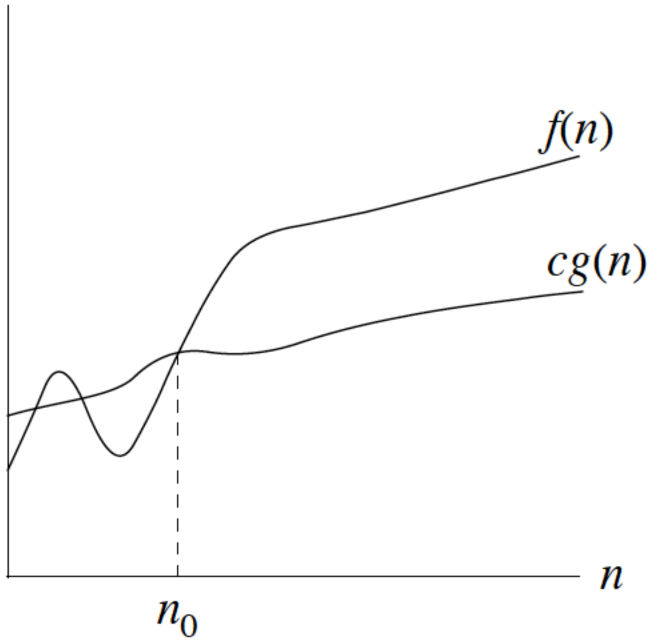
This was a **terrible** idea, but we're stuck with it!

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

If $f(n) \in O(g(n))$, we write $f(n) = O(g(n))$

Asymptotic notation

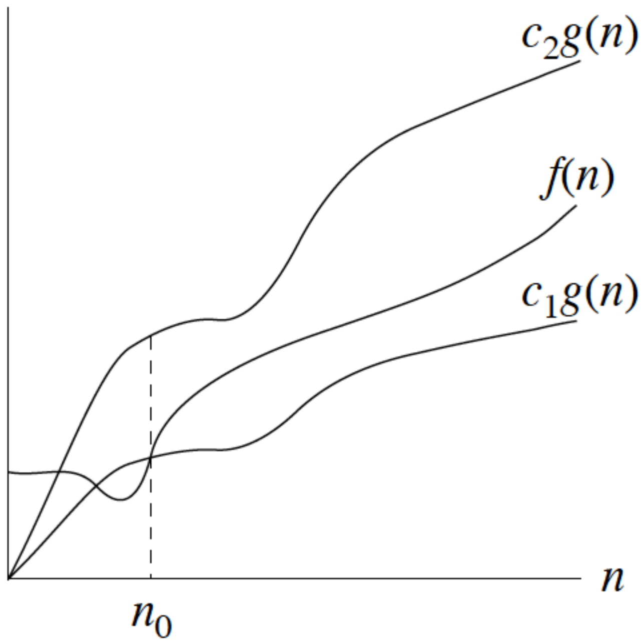
$$\Omega(g(n)) = \{ f(n) : \text{there exist positive constants } c \text{ and } n_0 \text{ such that } 0 \le cg(n) \le f(n) \text{ for all } n \ge n_0 \} .$$



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

Asymptotic notation

$$\Theta(g(n)) = \{ f(n) : \text{there exist positive constants } c_1, c_2, \text{ and } n_0 \text{ such that } 0 \le c_1 g(n) \le f(n) \le c_2 g(n) \text{ for all } n \ge n_0 \} .$$



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

Theorem 3.1

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

Asymptotic notation

$$o(g(n)) = \{f(n) : \text{for all constants } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le f(n) < cg(n) \text{ for all } n \ge n_0\}$$

$$: \lim_{n \to \infty} \frac{f(n)}{g(n)} = 0.$$

$$\omega(g(n)) = \{f(n) : \text{for all constants } c > 0, \text{ there exists a constant } n_0 > 0 \text{ such that } 0 \le cg(n) < f(n) \text{ for all } n \ge n_0\}$$

$$\lim_{n \to \infty} \frac{f(n)}{g(n)} = \infty.$$

Asymptotic notation in equations

When on right-hand side

$O(n^2)$ stands for some anonymous function in the set $O(n^2)$.

$2n^2 + 3n + 1 = 2n^2 + \Theta(n)$ means $2n^2 + 3n + 1 = 2n^2 + f(n)$ *for some* $f(n) \in \Theta(n)$. In particular, $f(n) = 3n + 1$.

When on left-hand side

No matter how the anonymous functions are chosen on the left-hand side, there is a way to choose the anonymous functions on the right-hand side to make the equation valid.

Interpret $2n^2 + \Theta(n) = \Theta(n^2)$ as meaning *for all* functions $f(n) \in \Theta(n)$, there exists a function $g(n) \in \Theta(n^2)$ such that $2n^2 + f(n) = g(n)$.

Can chain together:

$$\begin{aligned} 2n^2 + 3n + 1 &= 2n^2 + \Theta(n) \\ &= \Theta(n^2) . \end{aligned}$$

ASYMPTOTIC NOTATION: PROPERTIES

Transitivity:

$f(n) = \Theta(g(n))$ and $g(n) = \Theta(h(n)) \Rightarrow f(n) = \Theta(h(n))$.

Same for O, Ω, o, and ω.

Reflexivity:

$f(n) = \Theta(f(n))$.

Same for O and Ω.

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 $g(n) = \omega(f(n))$.

COMPARISON OF FUNCTIONS

omparisons:

$f(n)$ is ***asymptotically smaller*** than $g(n)$ if $f(n) = o(g(n))$.

$f(n)$ is ***asymptotically larger*** than $g(n)$ if $f(n) = \omega(g(n))$.

No trichotomy. Although intuitively, we can liken O to $\leq$, Ω to $\geq$, etc., unlike real numbers, where $a < b$, $a = b$, or $a > b$, we might not be able to compare functions.

Example: $n^{1+\sin n}$ and n, since $1 + \sin n$ oscillates between 0 and 2.

A Problem

Suppose that we wish to know which floors in an n-floor building are safe to drop eggs from, and which will cause the eggs to break on landing.

Assume that

- An egg that survives a fall can be used again (but a broken egg is discarded)
- The effect of a fall is the same for all eggs
- If an egg breaks when dropped, then it would break if dropped from a higher floor
- If an egg survives a fall then it would survive if dropped from a lower floor

You have k eggs available. What is the least number of egg-droppings that is guaranteed to determine the lowest floor from which it would break if dropped?

k = 1? k = infinity? Some smaller k?