Succinct Representation of Running Times

- We often calculate the running time as some complex expression, like $4n^2 + 7n + 8$.
- in the long run, that is asymptotically, what matters is the quadratic factor.
- It shall suffice to write $O(n^2)$ or $\Theta(n^2)$.

But...

- don't constants matter: isn't $2n^2$ better than $4n^2$?
  yes, but
  - advantage could be erased by simple optimization
  - depends on processor speed
- isn't n^2 better than $1,000,000n$?
  - only when $n \leq 1,000,000$
- isn't 2^n better than n^{10}?
  - only for really small n
  - exponential growth is bad

Big-O notation

We say that f is dominated by g, written $f \in O(g)$

$$\text{if } f(n) \leq g(n) \text{ for all } n \geq 0$$

or even (constants don't matter)

$$\text{if for some } c > 0 : f(n) \leq cg(n) \text{ for all } n \geq 0$$

or even (we only care about the "long run")

$$\text{if for some } c > 0, n_0 \geq 0 : f(n) \leq cg(n) \text{ for all } n \geq n_0$$

Properties of the O-relation

- reflexive: $f \in O(f)$ for all functions f
- transitive: if $f \in O(g)$ and $g \in O(h)$ then $f \in O(h)$
- "degree-preserving": $n^p \in O(n^q)$ iff $p \leq q$
- not total: we may have $f \notin O(g)$ and $g \notin O(f)$

Assume $f_1 \in O(g_1)$, $f_2 \in O(g_2)$. Then

- $f_1 f_2 \in O(g_1 g_2)$
- $f_1 + f_2 \in O(g)$ whenever $g_1 \leq g$, $g_2 \leq g$.

A sufficient condition for $f \in O(g)$:

$$\lim_{n\to\infty} \frac{f(n)}{g(n)} = c \text{ for some } c \text{ with } 0 \leq c < \infty$$

but this is not a necessary condition

Big-Omega and Big-Theta Notation

Big-Omega is dual to Big-O:

$$f \in \Omega(g) \text{ iff } g \in O(f)$$

In particular,

$$n^p \in \Omega(n^q) \text{ iff } p \geq q.$$

Big-Theta is the intersection of Big-O and Big-Omega:

$$f \in \Theta(g) \text{ iff } f \in O(g) \text{ and } f \in \Omega(g).$$

That is, there exists c_1, c_2 such that (from a certain point) $f(n)$ is "sandwiched" between $c_1 g(n)$ and $c_2 g(n)$. In particular ($a_p > 0$),

$$a_p n^p + \ldots + a_1 n + a_0 \in \Theta(n^q) \text{ iff } p = q$$

and also

$$\log_a(n) \in \Theta(\log_b(n)) \text{ for } a, b > 1$$

Worst/Best-Case Interpretation

Consider the running time of Insertion Sort:

- $\in O(n^2)$? yes
  - there exists $f \in O(n^2)$ such that for all n, and all input of size n, the running time is at most $f(n)$
- $\in O(n^3)$? yes (but potentially misleading)
- $\in O(n)$? yes for the best-case interpretation:
  - there exists $f \in O(n)$ such that for all n there exists input of size n with running time is at most $f(n)$
- $\in \Omega(n)$? yes
  - there exists $f \in \Omega(n)$ such that for all n, and all input of size n, the running time is at least $f(n)$
- $\in \Omega(n^2)$? yes for the worst-case interpretation:
  - there exists $f \in \Omega(n^2)$ such that for all n there exists input of size n with running time is at least $f(n)$
- $\in \Theta(n)$? for the best-case interpretation
- $\in \Theta(n^2)$? for the worst-case interpretation

Little-o, Little-omega

little-o means "grows much slower than":

$$f \in o(g) \text{ iff } \lim_{n\to\infty} \frac{f(n)}{g(n)} = 0$$

little-omega is dual to little-o:

$$f \in \omega(g) \text{ iff } g \in o(f)$$

Polynomials good, exponentials bad:

$$n^b \in o(a^n) \text{ for all } a > 1, b > 0$$

and hence logarithms grow slower than roots:

$$\log_{\mathbf{b}}(\mathbf{n}) \in \mathbf{o}(\sqrt[a]{n})\text{for all } a, b > 1$$

Overview

$n^p \in X(n^q)$ is equivalent to $p \; R \; q$ when

X	R
o	$<$
O	$\leq$
Θ	$=$
Ω	$\geq$
ω	$>$

If $f \in o(g)$ then

- $f \in O(g)$
- $f \notin \Theta(g)$

Does the converse relation hold?

- for "typical" functions, yes
- in general, no