Asymptotic Notations

Adapted from the CLRS book slides

OVERVIEW

How to evaluate the efficiency of an algorithm?

- A function from inputs to running times
- Needs a way to compare "sizes" of functions

Asymptotic Notation

- A way to describe behavior of functions in the limit.
- Describe growth of functions.
- Focus on what's important by abstracting away low-order terms and constant factors.

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

O-notation

O-notation characterizes an ***upper bound*** on the asympototic behavior of a function: it says that a function grows ***no faster*** than a certain rate. This rate is based on the highest order term.

For example:

$f(n) = 7n^3 + 100n^2 - 20n + 6$ is $O(n^3)$, since the highest order term is $7n^3$, and therefore the function grows no faster than n^3.

The function $f(n)$ is also $O(n^5)$, $O(n^6)$, and $O(n^c)$ for any constant $c \geq 3$.

Ω-notation

Ω-notation characterizes a ***lower bound*** on the asymptotic behavior of a function.

For example:

$f(n) = 7n^3 + 100n^2 - 20n + 6$ is $\Omega(n^3)$, since the highest-order term, n^3, grows at least as fast as n^3.

The function $f(n)$ is also $\Omega(n^2)$, $\Omega(n)$ and $\Omega(n^c)$ for any constant $c \leq 3$.

Θ-notation

Θ-notation characterizes a ***tight bound*** on the asympototic behavior of a function: it says that a function grows ***precisely*** at a certain rate, again based on the highest-order term.

If a function is is both $O(f(n))$ and $\Omega(f(n))$, then a function is $\Theta(f(n))$.