Asymptotic Notation

Introduction

Asymptotic notation is notation that describes the approximate limiting behavior of functions, and includes "big O" and "little o" notation, Ω notation, and Θ notation,

The complexity of a computation is a measure of the resources used to make the computation. There is time complexity, a measure of the time used, and space complexity, a measure of how much memory is needed.

Since machines operate a different speeds, a measure of computation time in seconds, minutes, or hours, is dependent on the machine, and is thus not a true measure of the complexity of the algorithm used. Asymptotic notation is used to abstract from the technology of the machine. For example, Euclid's algorithm for finding the greatest common divisor of two integers, worked on a modern computer, takes less time than on an older computer, or by hand. But the true efficiency of the algorithm has not changed.

Complexity of Functions

For now, we consider only functions of non-negative integers to non-negative integers. If we write $\sqrt{13}$, for example, we mean an integer close to the real square root of 13, and if we write $13/2$, we mean 6. When we write $\log_2 n$, we mean an integer which is close to the real value.

O Notation. If we write $f(n) = O(g(n))$, where f and g are functions, we mean that eventually $f(n)$ does not exceed some constant multiple of $g(n)$. More formally, there exist positive constants C and N such that $f(n) \leq C\,g(n)$ for all $n > N$.

- $5n + 17 = O(n)$. Proof: Pick $C = 10$ and $N = 20$. Then $2n + 17 \leq 5n$ for all $n \geq 20$.
- $f(n) = n^2 \neq O(n)$. Proof: Let C and N be constants. Suppose $f(n) = n^2 \leq Cn$ for all $n \geq N$. Let $n = \max\{N, C+1\}$. Then $n \geq N$, but $f(n) = n^2 = n\,n \geq (C+1)\,n > C\,n$, contradiction.

Ω Notation. The inverse of O-notation. $f(n) = \Omega(g(n))$ if and only if $g(n) = O(f(n))$. An alternative definition is that $f(n) = \Omega(g(n))$ means that there is some positive real number C and integer N such that $f(n) \geq C\,g(n)$ for all $n \geq N$.

Θ Notation. $f(n) = \Theta(g(n))$ means that both $f(n) = O(g(n))$ and $f(n) = \Omega(g(n))$.

Example. If $b > 1$ is a real constant, then $\log_b n = \Theta(\log_2 n)$. Thus, if we write $O(\log n)$, $\Omega(\log n)$, or $\Theta(\log n)$, the base doesn't matter, so we do not write it.