

Big-O Complexity

Computing

How an algorithm's cost grows, from constant to catastrophic.

Operations needed as the input n grows. The right-hand column is $n = 1,000,000$.

Class	Name	At $n = 1\text{M}$
$O(1)$	Constant	1
$O(\log n)$	Logarithmic	20
$O(n)$	Linear	1,000,000
$O(n \log n)$	Linearithmic	20,000,000
$O(n^2)$	Quadratic	10^{12}
$O(2^n)$	Exponential	beyond counting
$O(n!)$	Factorial	beyond counting

Constants are dropped: $O(2n)$ is $O(n)$. That matters less than the class, and only the class scales.