

Time Complexity of Algorithms

Big-O Notation	Name	Description	Example Algorithms
$O(1)$	Constant Time	Execution time is constant, regardless of input size.	Array index access, Hash table lookup
$O(\log n)$	Logarithmic Time	Time increases logarithmically as input size grows.	Binary Search, BST lookup
$O(n)$	Linear Time	Execution time grows directly with input size.	Linear Search, Traversing an array
$O(n \log n)$	Log-Linear Time	Slightly worse than linear, common in sorting.	Merge Sort, Quick Sort (avg case)
$O(n^2)$	Quadratic Time	Execution time grows quadratically with input size.	Bubble Sort, Selection Sort
$O(n^3)$	Cubic Time	Execution time grows cubically with input size.	Matrix Multiplication (Naïve)
$O(2^n)$	Exponential Time	Execution time doubles with each input increase.	Recursive Fibonacci, Subset Sum
$O(n!)$	Factorial Time	Execution time grows factorially, extremely slow.	TSP (Brute force), Permutations