Big O Notation

Notation	Description	Use Case
O(1)	O(1) is constant time. No matter how much data, it executes in constant time	Hashing
O(N)	As the data set grows, the less effective this algorithm is.	Linear search
$O(N^2)$	This algorithm can be seen in a nested loop. It grows proportionally to the data set	Bubble sort
O(log N)	Very efficient with large data sets. It halves itself repeatedly	Binary search
O(n log N)	Dividing data sets	Quick sort

Algorithm	Complexity
Linear time	O(n)
Quadratic time	$O(n^2)$
Logarithmic time	O(log n)
Constant time	O(1)
Exponential time	O(2^n)