Big O Summary (Arranged from Fastest to Slowest)

Time Complexity	Description	Example
O(1) - Constant	Time taken remains same regardless of input size	Access element in array
O(log N) - Logarithmic	Time taken increases logarithmically as the input size grow. Operations are typically halved at each step. Time increases linearly as N goes up exponential	Binary search in a sorted array
O(N) - Linear	Time grow when input grow	Find an item in unsorted list
O(N log N) - Linearithmic	Time taken increases in a linearithmic manner, often seen in divide and conquer algorithms	Merge sort or quicksort.
O(N^2) Quadratic	Time taken increases quadratically as input increases. (Nested loop)	Bubble sort or Selection sort.
O(2^N) Exponential	Time take doubles with each addition to N, leads to grow ~~rapid~~ execution time	Finding all subset of a set
O(N!) Factorial	Time taken increases factorially with each increase in input size, leading to extremes slow exec time	Solving the travelling salesman problem exhaustively.