Operation	Time Complexity	Space Complexity	Real-Life Analogy	Example
Access in Array	O(1)	O(n)	Grabbing a book by its shelf number	arr[5]
Linear Search	O(n)	O(n)	Scanning a crowd to find a friend	Looping through a list
Binary Search	O(log n)	O(n)	Opening a dictionary from the middle, narrowing down	On a sorted array
Insert/Delete (Array)	O(n)	O(n)	Adding a book to the start of a shelf	arr.insert(0, x)
Insert/Delete (Linked List)	O(n) (at any position)	O(n)	Adding/removing sticky notes from a notebook	deleteAt(index) / insertAt(x)
Stack/Queue (Push/Pop)	O(1)	O(n)	Stack of plates or people in a queue	stack.push(x) / queue.pop()
HashMap (Insert/Find)	O(1) avg / O(n) worst	O(n)	Like a dictionary lookup	map[key] = value
DFS/BFS (Graph)	O(V + E)	O(V + E)	Exploring rooms (DFS) or floors (BFS) in a building	Pathfinding
Merge Sort	O(n log n)	O(n)	Cutting and merging paper strips	Recursive sorting
Quick Sort (avg/worst)	O(n log n) / $O(n^2)$	O(log n)	Picking a pivot and sorting around it	Recursive
Heap Insert/Delete	O(log n)	O(n)	Maintaining a leaderboard	MinHeap / MaxHeap
Trie Insert/Search	O(L) (L = length of word)	O(n × L)	Auto-complete in search engines	trie.insert("cat")
Backtracking	O(2^n)	Varies	Trying every possible outfit before deciding	N-Queens, Subsets