Time Complexity Tips and Examples

This document provides an overview of time complexity tricks and tips along with examples for each concept to help in efficient programming.

Time Complexity Tip	Example
O(1): Constant Time	Accessing an element in an array by index: arr[i]
O(log n): Logarithmic Time	Binary Search on a sorted array
O(n): Linear Time	Iterating through an array: for i in arr
O(n log n): Divide & Conquer	Merge Sort or Quick Sort
O(n^2): Quadratic Time	Nested loops like Bubble Sort: for i in range(n): for j in range(n)
O(2^n): Exponential Time	Solving the Traveling Salesman Problem (TSP) with brute force
Hash Maps for Efficiency	Using a hash map to store word frequencies
Memoization in DP	Storing Fibonacci values to avoid redundant calculations
Binary Search Trees (BST)	Insertions and lookups in a BST are O(log n)
Precomputation	Precomputing factorials to avoid recalculating them repeatedly
Amortized Time Complexity	Appends in dynamic arrays (O(1) on average despite resizing)