Introduction to Algorithms

ddas.tech/introduction-to-algorithms/

October 2, 2022

Created By: Debasis Das (Oct 2022)

Table of Contents

Big O Notation

Big O is the language and Metrics used to describe the efficiency of Algorithms.

Time Complexity: Is the computational complexity that describes the amount of time it takes to run the algorithm. Consider the worst case time complexity before shipping your code.

Time Complexity – Best Case, Worst Case and Expected Case:

Runtimes of Algorithms can be described using Best case, Worst case and Expected Case.Taking the example of a **quick sort** algorithm. Quick sort picks a random element as a pivot and then swaps elements in the array such that the elements less than the pivot appear before elements greater than the pivot. This will give a partial sort, Then it is recursively sorts the left and right sides using a similar process.

For Quick Sort,

- Best Case is O(n) – If all elements are equal.
- Worst Case is O(n^2) – If the pivot is repeatedly the biggest element in the array.
- Expected Case is O(n log n)

Space Complexity: Time is not the only thing that matters in an algorithm, we need to care about the amount of memory or space required by the algorithm. Space complexity is a parallel concept to time complexity. If we need a one dimensional array space complexity is O(N) , if we need a two dimensional array of N*N dimensions then the space complexity is O(N^2)

Run Time Analysis – Is used for classifying algorithms according to how their run time and space requirements grows as input grows.

eg: O(1) does not change with the change in the size of the data structure.

The **letter O** is used because the growth of rate of a function is also referred to as the **Order** of the function

Big O Notation – Time Complexity Table

Time Complexity	Big O Notation	Applications, eg
Constant Time	O(1)	– Finding the median value in a sorted Array – Stack Push | Queue- Enqueue – Stack POP | Queue- Dequeue
Logarithmic	O(Log n)	Finding an item in a sorted array with binary search or a balanced search tree
Linear	O(n)	Finding an item in an unsorted list or unsorted array
Log Linear	O(n log n)	Fast Fourier Transform Comparison Sort Heap Sort Merge Sort
Quadratic	O(n^2)	Bubble Sort Selection Sort Insertion Sort
Cubic	O(n^3)	Naive multiplication of two n*n matrix
Exponential	O(c^n)	Traveling Salesman Problem using Dynamic Programming Matrix Chain Multiplication using brute force
Factorial	O(n!)	Traveling Salesman Problem using Brute Force

Big O Notation – Time Complexity.