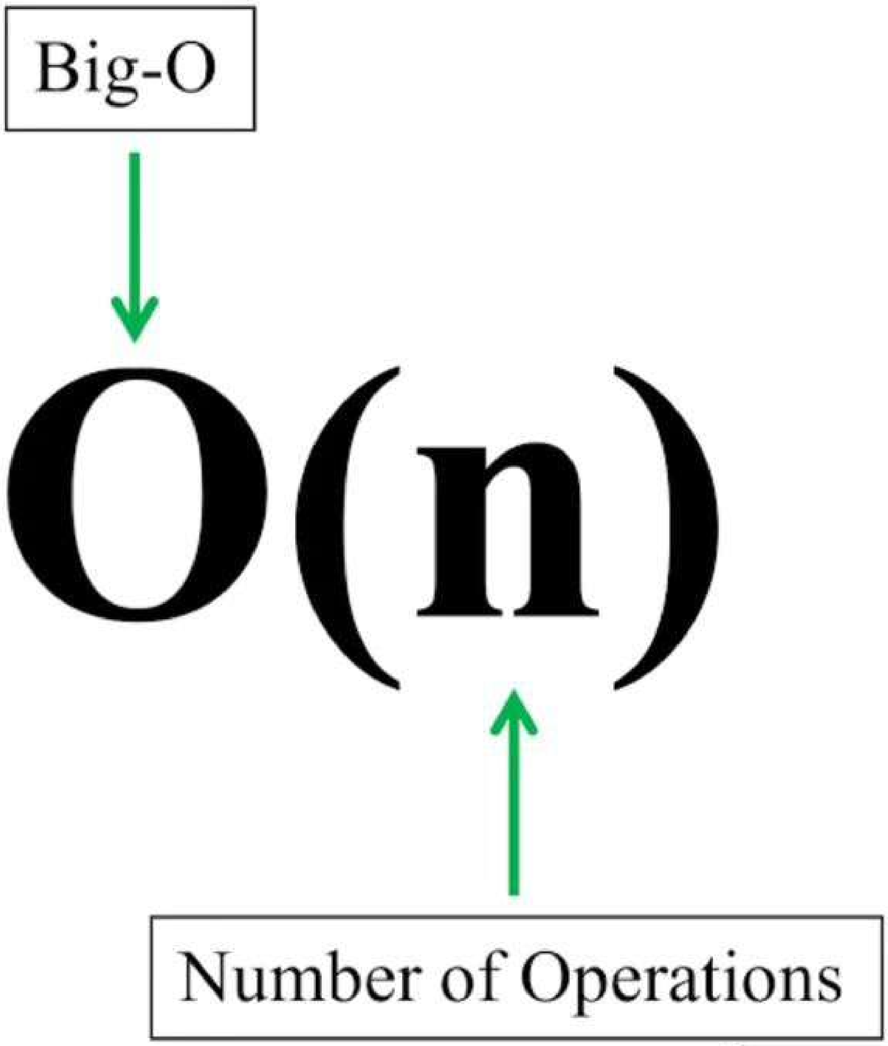
BIG O
Notation

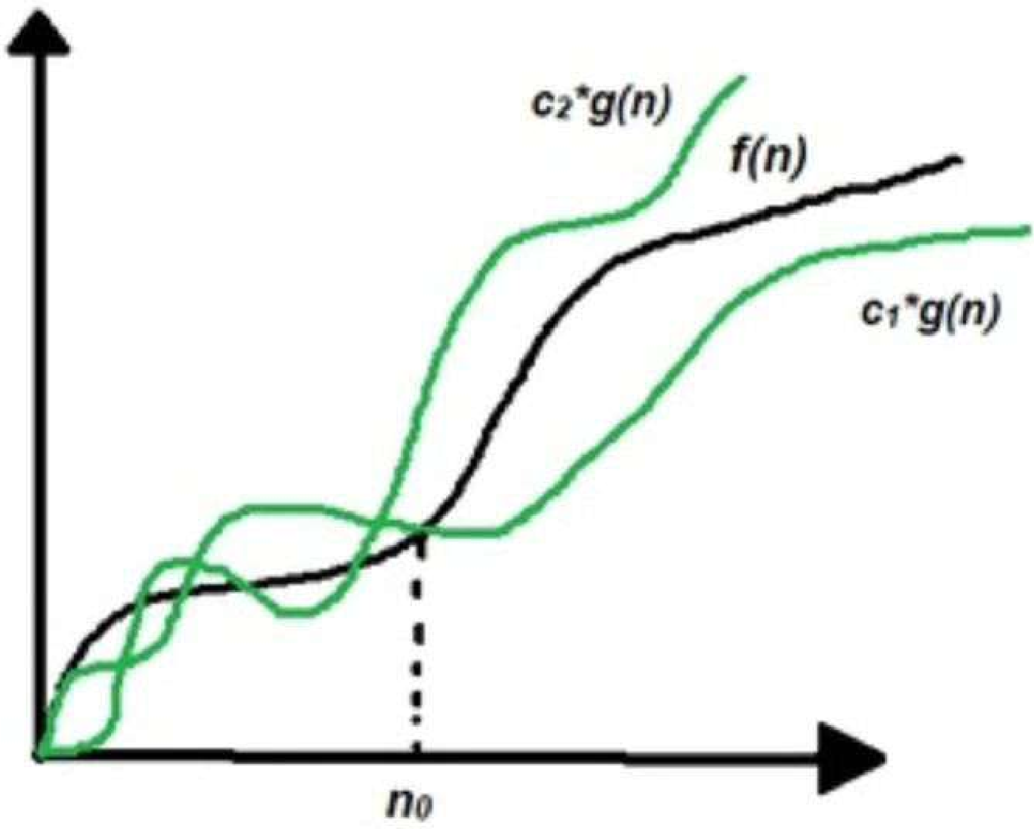


DM for Project Help

Swipe >>>

What is Asymptotic Notations?

Asymptotic Notations are the mathematical Notations used when the input tends towards a particular or a limiting value while describing the running duration of an algorithm.

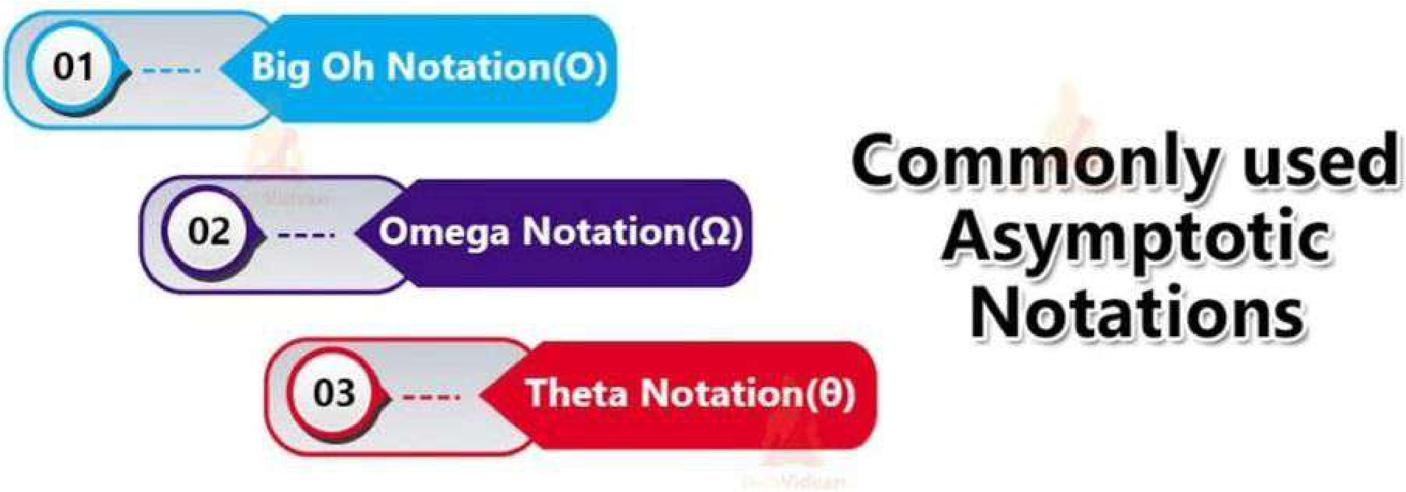


DM for Project Help

Swipe >>>

Types of Asymptotic Notations?

There are three Notations are commonly used. The three asymptotic Notations below are generally used to indicate algorithms' time complexity.



BIG - O Notation

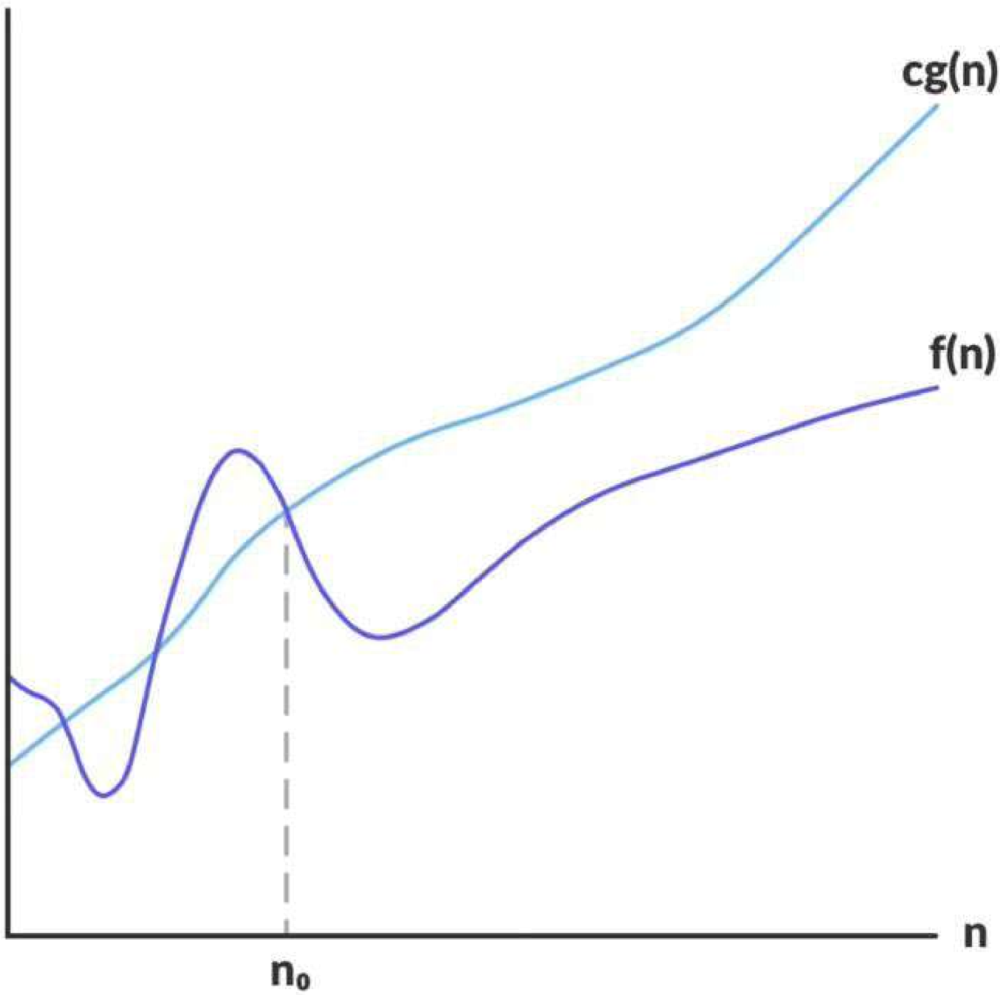
BIG - O Notation is used to calculate how scalable and efficient an algorithm is relative to its input considering the Time and Space Complexity

$$O(n)$$

DM for Project Help

Swipe >>>

- "Order 1" is a constant time/method: (1)
- "Order N" is a linear function/method: O(N)
- The "order N squared" is a quadratic-time feature/method: E. (N 2)



$f(n) = O(g(n))$

DM for Project Help

Swipe >>>

Time Complexity

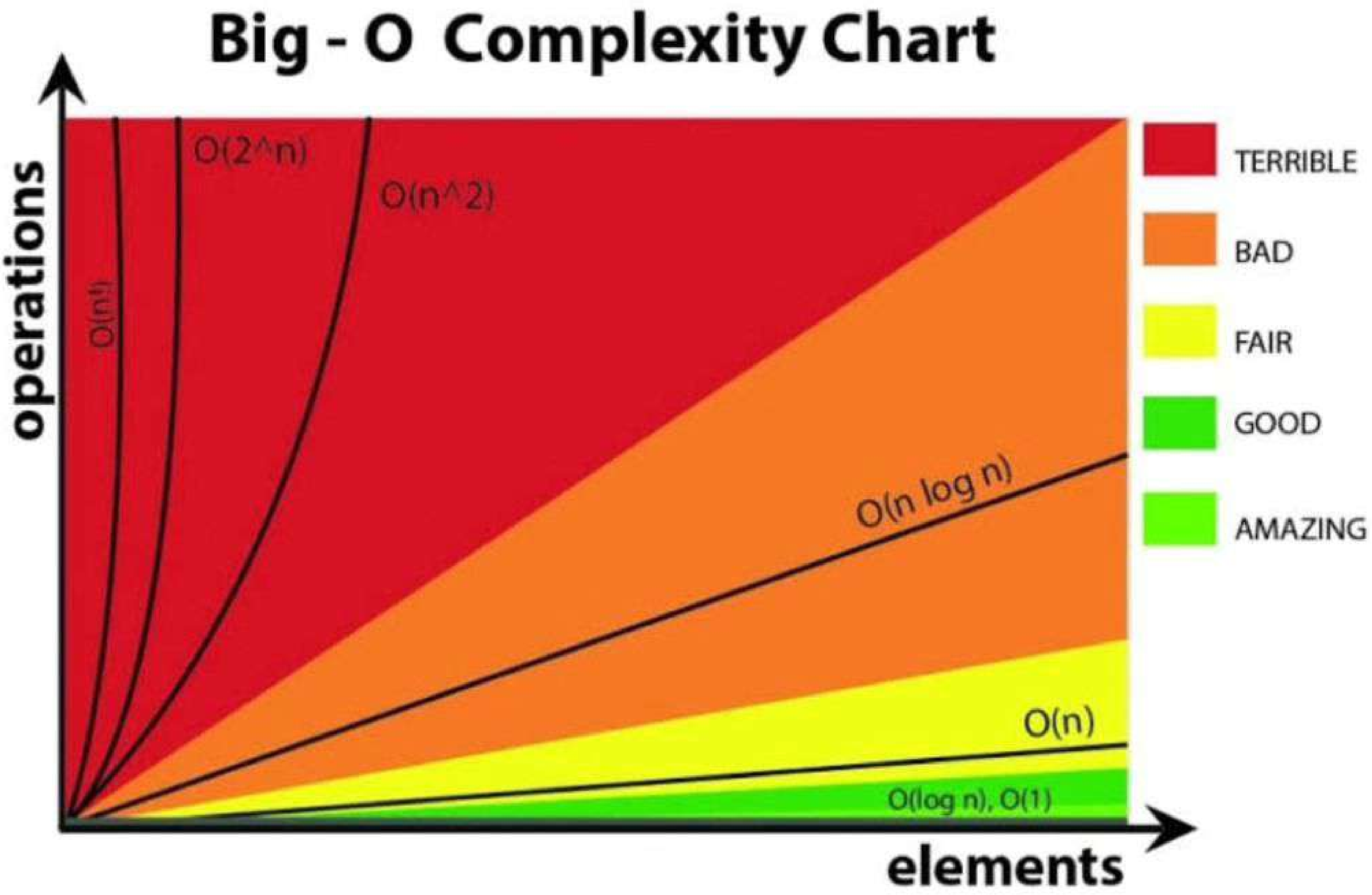
How much time an algorithm takes to run completely ?

Space Complexity

How much extra space does an algorithm require in the process ?

Which are the Types of Big-O Notation?

Notation	Type	Examples	Description
O(1)	Constant	Hash table access	Remains constant regardless of the size of the data set
O(log n)	Logarithmic	Binary search of a sorted table	Increases by a constant. If n doubles, the time to perform increases by a constant, smaller than n amount
O(<n)	Sublinear	Search using parallel processing	Performs at less than linear and more than logarithmic levels
O(n)	Linear	Finding an item in an unsorted list	Increases in proportion to n. If n doubles, the time to perform doubles
O(n log(n))	n log(n)	Quicksort, Merge Sort	Increases at a multiple of a constant
$O(n^2)$	Quadratic	Bubble sort	Increases in proportion to the product of n*n
$O(c^n)$	Exponential	Travelling salesman problem solved using dynamic programming	Increases based on the exponent n of a constant c
O(n!)	Factorial	Travelling salesman problem solved using brute force	Increases in proportion to the product of all numbers included (e.g., 1*2*3*4...)



DM for Project Help

Swipe >>>

Example:

Let's say you are running a website that sells products and you want to optimize its search algorithm to make it more efficient. You have two different **algorithms** to choose from: **Algorithm A** and **Algorithm B.**

Algorithm A takes **O(n)** time to search for a product, where **n** is the number of products in your database. This means that as the number of products increases, the time it takes to search for a product increases linearly.

Swipe >>>

Algorithm B takes **O(log n)** time to search for a product, which means that as the number of products increases, the time it takes to search for a product increases logarithmically. This algorithm is generally faster than Algorithm A for large datasets.

In this case, you might choose Algorithm B because it scales better as your database grows, and it will be more efficient in the long run.