

Data Structures

A data structure organizes data for efficient use and is essential for creating powerful algorithms. Another reason for using data structures is to manage and organize data naturally. To understand the performance that data structures provide, we need to look at the wild world of computational complexity.

The abstract data type defines how a data structure should behave and what methods it should have. However, it does not provide the details surrounding how those methods are implemented. Big O notation only cares about what happens when input becomes arbitrarily large, ignoring things like constants and multiplicative factors. Almost any mathematical expression containing n can be wrapped around a Big O. If the algorithm takes logarithmic or quadratic/cubic time, then it's represented as Big O of a log event.

Constant Time Algorithms

Both of the following algorithms run in constant time relative to the input size because they are independent of n . As the input size grows infinitely large, the loop still runs for the same amount of time.

- Algorithm 1
- Algorithm 2

Binary Search Algorithm

A classic algorithm for binary search has a logarithmic time complexity. This algorithm starts by creating two pointers: one at the beginning and one at the end of the array. Then, it selects a midpoint between the two pointers and checks if the value being searched for is found at the