Data Structures & Algorithms from a GOOGLE Engineer!

Will be doing this in Java

**DS** ? It is a way of organizing data so that it can be used effectively.

They are essential ingredients in creating fast and powerful algorithms. They make code cleaner and easier. Help us manage our data.

**Abstract data types (ADT) vs DS ?**

An ADT is an abstraction of a data structure which provides only the interface to which a DS must adhere to. The interface doesn’t give any necessary detail about how something is implemented or working.

**ADT ------------------------- Implementation (DS)**

List ---------------- Dynamic Array , Linked List

Queue --------------- Linked List based , Array based , Stack based Queues

Map ------------------ Tree Map , HashMap

**Computational Complexity Analysis :**

As programmers , we often ask ourselves how long this algorithm will take to finish and how much space would it consume ?

**Big - O Notation** gives an upper bound of the complexity in the worst case , helping to quantify performance as the input size became arbitrarily large.

(n) – the size of the input

1. Constant Time – O(1)
2. Logarithmic Time – O(log(n))
3. Linear Time – O(n)
4. Linearithmic Time – O(nlog(n))
5. Quadric Time – O(n^2)
6. Cubic Time – O(n^3)
7. Exponential Time – O(b ^ n) , b > 1
8. Factorial Time – O(n!)

**Big-O Properties** :

O(n + c) = O(n)

O(nc) = O(n)

a := 1

b := 2

c := a + 5b

This will run in constant time.

**Static and Dynamic Arrays :**

Arrays are a fundamental building block of all other data structures. With arrays and pointers alone , we can look everywhere , we can construct any sort of data.

**Arrays ?**

A static array is a fixed length container containing n elements indexable from the range [0 , n-1].

These arrays are a contiguous chunks of memory.

Uses :

1. Storing and accessing sequential data.
2. Temporarily storing data.
3. Used by IO routines as buffers.
4. Lookup tables and inverse lookup tables.
5. Can be used to return multiple values from a function.
6. Used in dynamic programming to cache answers to subproblems.

**STATIC ARRAY DYNAMIC ARRAY**

Access O(1) O(1)

Search O(n) O(n)

Insertion N/A O(n)

Appending N/A O(1)

Deletion N/A O(n)

**Dynamic Arrays ?**

Dynamic arrays can grow and shrink in size.

**How can we implement a Dynamic Array using Static array ?**

1. Create a static array with an initial capacity.
2. Add elements to the underlying static array , keeping track of the number of the elements.
3. If adding another element will exceed the capacity , then create a new static array with twice the capacity and copy the original elements into it.