Lecture 1:

Data Structures...what & why?

"I will, in fact, claim that the difference between a bad programmer and a good one is whether he considers his code or his data structures more important. Bad programmers worry about the code. Good programmers worry about data structures and their relationships."

– Linus Torvalds

Data Structure:

A data structure is a way we store and organize our data.

For example: Think about organizing books in a room:

- We can keep them on a shelf or,
- make a stack of them on a table or,
- just put them randomly anywhere in the room.

i.e. we have different options to organize books in a room or we can say different structures to keep books.

In a similar manner, in computers also we can organize our data in the way we want and these different ways of organizing data are different data structures.

Now, from our example of organizing books, one will try to organize the books in such a way that say frequently used books can be kept on shelf to access them with ease instead of keeping them at the bottom of the pile of books.

So,

Data structures are a way of organizing and storing data so that operations can be performed efficiently.

Five Fundamental Behaviors (access, insert, delete, find & sort)

Here operations are like accessing, inserting, deleting, finding and sorting the data which are basic operations that one can perform using data structures.

Importance of Data Structure and Algorithm:

When we study data structure and algorithm, we are concerned ***about space (RAM) and processing (CPU) efficiency***.

Both *processing power and memory are finite*. So, we need to **keep things optimized** so that it takes least amount of space and processing time.

So, we can also say Data structure is a way of organizing the items in terms of memory, and also the way of accessing each item through some defined logic.

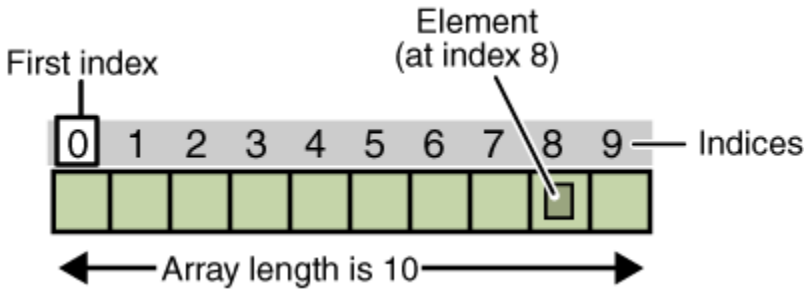
Data Structure Types:

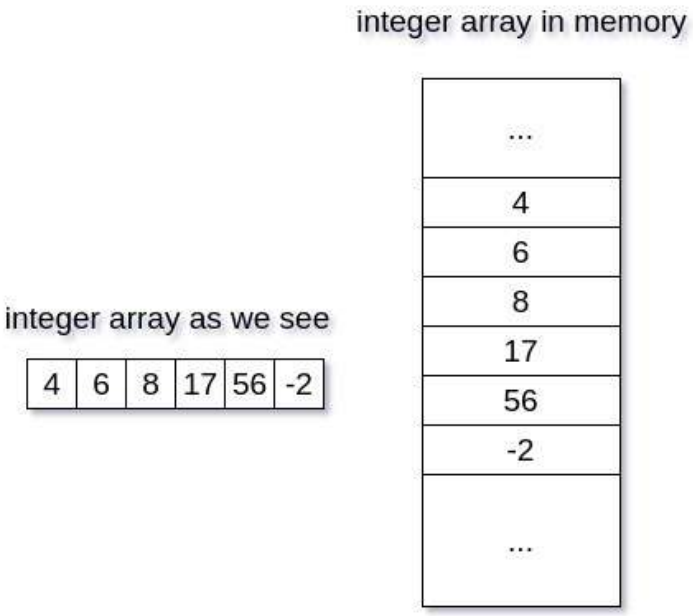
1. **Linear Data Structure:**
   - Elements of linear data structure are accessed in a sequential manner, however the elements can be stored in these data structures in any order. Examples of linear data structure are: **Linked List, Stack, Queue and Array.**
2. **Non-Linear Data Structure:**
   - Elements of non-linear data structures are stored and accessed in non-linear order. Examples of non-linear data structure are: **Tree and Graph.**

Array:

An array is an ordered collection of items, where each item inside the array has an index.

An array is fundamentally a list of similar values. Elements of an array are stored in contiguous memory blocks. All the elements have to have the same data types. Arrays have a fixed length i.e. immutable. They can be created initially at any size. But once the array is created, you can't add or remove elements from it.





An array usually occupies a block of memory space, one cannot simple double or triple the size of an array when additional space is required → **arrays are called dense lists and are said to be static data structure.**

Works Cited (htt2) (htt3) (Lipschutz)

(n.d.). Retrieved from https://medium.com/omarelgabrys-blog/diving-into-data-structures-6bc71b2e8f92

(n.d.). Retrieved from https://medium.com/@tssovi/introduction-to-data-structures-7db7a0d5ae0

(n.d.). Retrieved from https://medium.com/swlh/introduction-to-data-structures-9134b7d064a6

https://medium.com/swlh/data-structures-101-e18fc33579fa. (n.d.). Retrieved from https://medium.com/swlh/data-structures-101-e18fc33579fa

Lipschutz, S. *Data Structures (Schaum's Outlines).* The McGraw Hill Companies.