[redacted] (/)

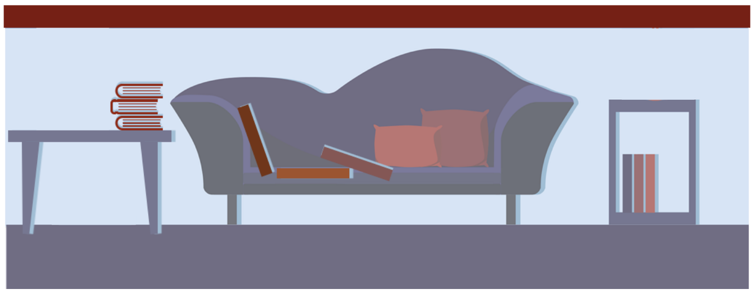
Introduction to Data Structures

Data Structures and Algorithms (or DSA) is the most important course of any Computer Science program. In this course, we will deal with different data structures, their applications, running times, etc. If you don't know how to analyze the running time of code, you can read the first 7 chapters of the Algorithm Course (https://www.codesdope.com/course/algorithms-introduction/). It is also recommended that you first read those chapters before proceeding with this course.

In this very first chapter of data structures, we will focus on learning what basically is a data structure and why do we need it. So, let's start.

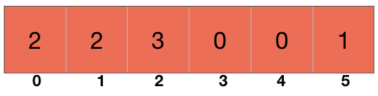
What is a data structure?

A data structure is a way we store and organize our data. For example, think about organizing books in a room, we can keep those books on a shelf, or make a stack of them on a table or even just put them randomly anywhere in the room.



Thus, we have different options to organize books in a room or in different words, we have different structures to keep books. In computers also, we have a similar scenario i.e., we can organize our data in the way we want and these different ways of organizing data are different data structures.

For example, an array is a type of data structure which we learn while learning basic programming languages. It is the most basic data structure and stores different data at different indices.

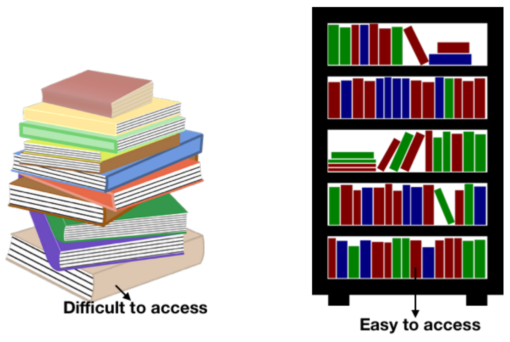


An Array

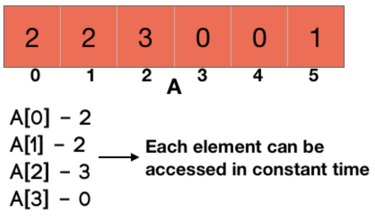
There are many different data structures which are generally used. Many programming languages also provide pre-built libraries for many data structure. But in this course, every discussed data structure is made from scratch.

Do we really need to worry about how our data is stored?

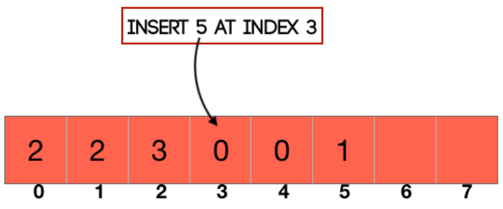
One can keep a frequently used book at the bottom of the pile of books and can access it with a little difficulty but it would make a lot more sense to keep frequently used books on the shelf to access them with ease.



In computers also, the choice of the data structure depends upon the task we are going to perform. For example, if we have a constant number of data and accessing the data in the least time is our priority, then an array is a suitable data structure because it can return the data at an index in constant time ($O(1)$).



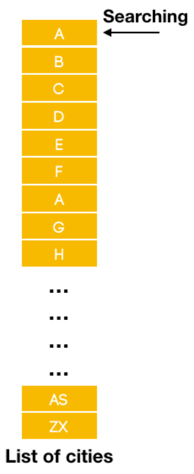
But imagine a task in which we need to frequently insert some new data between two data. In that case, using an array will lead to shifting the elements of the array or even making a new array of different size if the array is not large enough.



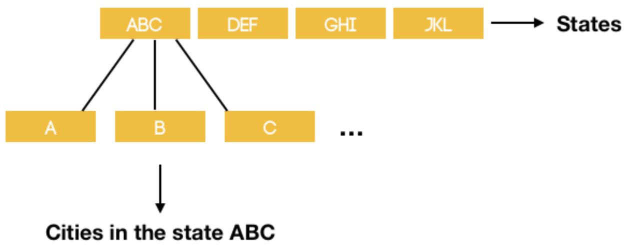
So, a data structure in which the task of inserting some new data between two data is done in the least time would be suitable for this purpose.

The point is that we can complete a task using any data structure but a suitable data structure for a task not only reduces the programmer's effort but also saves a lot of computational time and space.

For example, imagine searching for a city in the list of all the cities of a country. If the desired city is at the last of the list, we will end up iterating over the entire list.



But if we organize all the cities under the state in which they lie and we know the state, it would be a much quicker process to search the city.



Can’t I just use libraries instead of making a data structure from scratch?

The first point is that you need to at least understand the working of the data structure even to use a library. So, assuming that a person has an understanding of the data structure being used and the library provides exactly what the person needs, of course, a library can be used.

Even though we can use a library for simpler data structures but we often need a more complex data structure which is made using simpler data structures and existing libraries of them doesn’t always provide exactly what we need and we end up writing our own data structure from scratch. Sometimes this also happens with simpler data structures and we also make them from scratch to suit our need.

Should I also be concerned with the choice of language for the implementation of data structure?

(/add_ques

The implementation in a language like C is done with the help of structure, pointer, etc. Whereas in an objected oriented language like Java, it is done with classes and objects and the idea remains the same as long as the language is an object-oriented one. So, the implementation will change with the "**type**" of the language we are using.

In this course, we are going to implement every data structure in three different languages - C/C++, Java and Python, you can proceed with the language you know.

What this course teaches me?

In this course, you will learn the basic concepts of different data structures, their applications and their implementation in different languages. We will also focus on the running time of different processes like inserting data, searching data, etc. in a data structure. At the end of this course, you will have knowledge of different data structures and you can use this knowledge to create a new data structure or modify an existing one according to your need.

(/course/data-structures-linked-lists/) NEXT