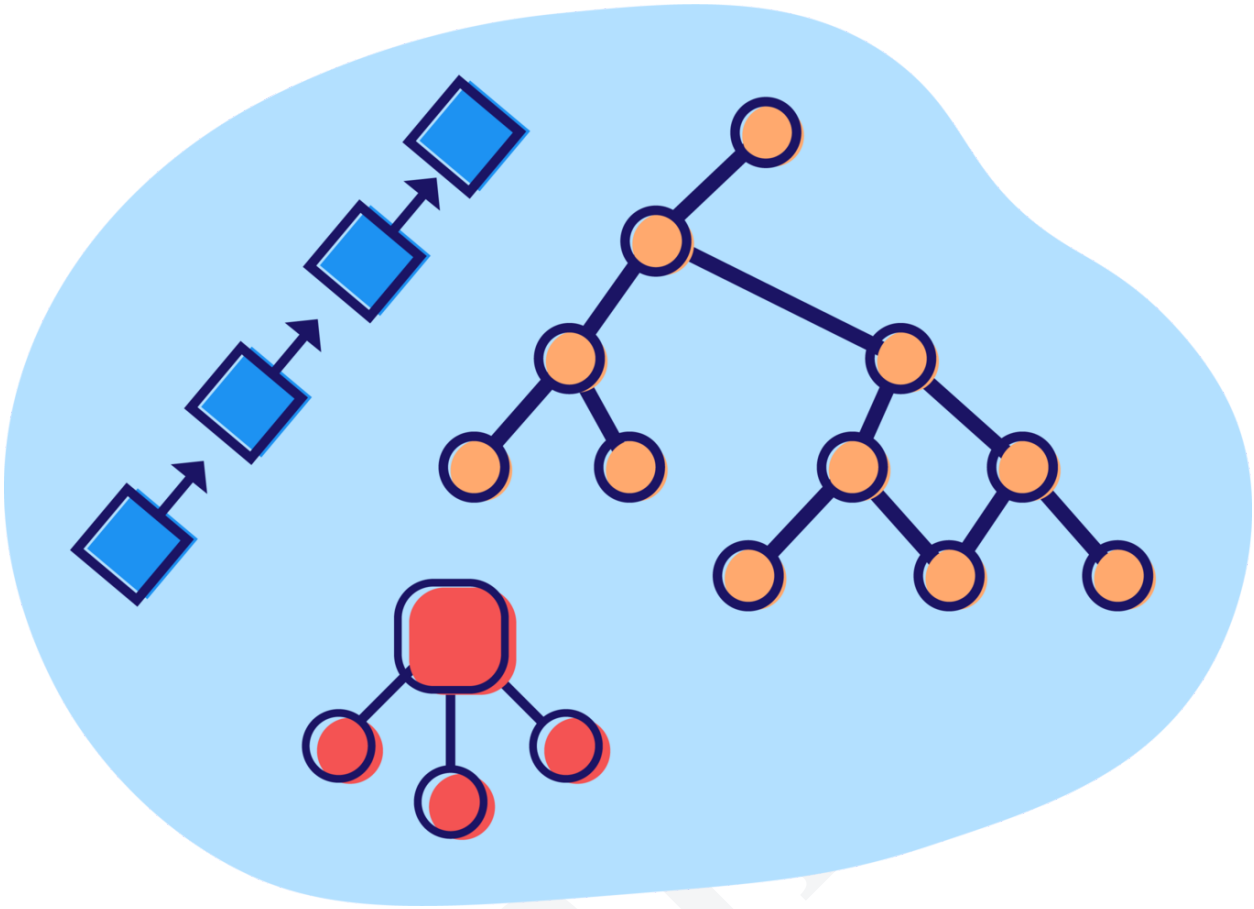

Data Structures & Algorithms

Chapter - 01

Basics Of Data Structures & Algorithms

Rakin Mohammad Sifullah
Computer Science Engineer

Overview

In this chapter, we will cover the basic understanding of Data Structures & Algorithms.

We will cover -

1. What is Data Structure?
2. Why should we need to know Data Structures and Algorithms?
3. Data Structures & Algorithms application.
4. What are the characteristics of Data Structures & Algorithms?
5. Execution time cases
6. Terminologies
7. Prerequisites

What is Data Structure?

Data Structure can be defined as the group of data elements that provides an efficient way of storing and organizing data in the computer so that it can be used efficiently. Some examples of Data Structures are arrays, Linked List, Stack, Queue, etc. Data Structures are widely used in almost every aspect of Computer Science i.e. Operating systems, Compiler Design, Artificial intelligence, Graphics, and many more.

Data Structures are the main part of many computer science algorithms as they enable the programmers to handle the data in an efficient way. It plays a vital role in enhancing the performance of the software or a program as the main function of the software is to store and retrieve the user's data as fast as possible.

Why should we need to know Data Structures and Algorithms?

As applications are getting complex and data-rich, there are three common problems that applications face nowadays.

- Data Search − Consider an inventory of 1 million(106) items of a store. If the application is to search for an item, it has to search an item in 1 million(106) items every time slowing down the search. As data grows, the search will become slower.
- Processor speed − Processor speed although very high, falls limited if the data grows to billion records.
- Multiple requests − As thousands of users can search data simultaneously on a web server, even the fast server fails while searching the data.

To solve the above-mentioned problems, data structures come to the rescue. Data can be organized in a data structure in such a way that all items may not be required to be searched, and the required data can be searched almost instantly.

Data Structures & Algorithms application

An algorithm is a step-by-step procedure, which defines a set of instructions to be executed in a certain order to get the desired output. Algorithms are generally created independent of underlying languages, i.e. an algorithm can be implemented in more than one programming language (C, C++, JAVA, Python, etc.).

From the data structure point of view, the following are some important categories of algorithms −

- Search − Algorithm to search an item in a data structure.
- Sort − Algorithm to sort items in a certain order.
- Insert − Algorithm to insert an item in a data structure.
- Update − Algorithm to update an existing item in a data structure.
- Delete − Algorithm to delete an existing item from a data structure.

The following computer problems can be solved using Data Structures −

- Fibonacci number series
- Knapsack problem
- Tower of Hanoi
- All pair shortest path by Floyd-Warshall
- Shortest path by Dijkstra
- Project scheduling

What are the characteristics of Data Structures & Algorithms?

- Correctness − Data structure implementation should implement its interface correctly.
- Time Complexity − The running time or the execution time of operations of the data structure must be as small as possible.
- Space Complexity − Memory usage of a data structure operation should be as little as possible.

Execution time cases

There are three cases that are usually used to compare various data structures' execution times in a relative manner.

- Worst Case − This is the scenario where a particular data structure operation takes the maximum time it can take. If an operation's worst-case time is ƒ(n) then this operation will not take more than ƒ(n) time where ƒ(n) represents the function of n.
- Average Case − This is the scene depicting the average execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then m operations will take mƒ(n) time.
- Best Case − This is the scene depicting the least possible execution time of an operation of a data structure. If an operation takes ƒ(n) time in execution, then the actual operation may take time as the random number which would be maximum as ƒ(n).

Terminologies

- Data – Data are values or sets of values.
- Data Item – Data item refers to a single unit of values.
- Group Items – Data items that are divided into sub-items are called Group Items.
- Elementary Items – Data items that cannot be divided are called Elementary Items.
- Attribute and Entity – An entity is that which contains certain attributes or properties, which may be assigned values.
- Entity Set – Entities of similar attributes form an entity set.
- Field – A field is a single elementary unit of information representing an attribute of an entity.
- Record – A record is a collection of field values of a given entity.
- File – A file is a collection of records of the entities in a given entity set.

Prerequisites

Before proceeding with Data Structures & Algorithms should have a basic understanding of C programming language, text editor, and execution of programs, etc.