DAILY DSA | DAY-29 | Summary |
-GOPALKRISHNA A

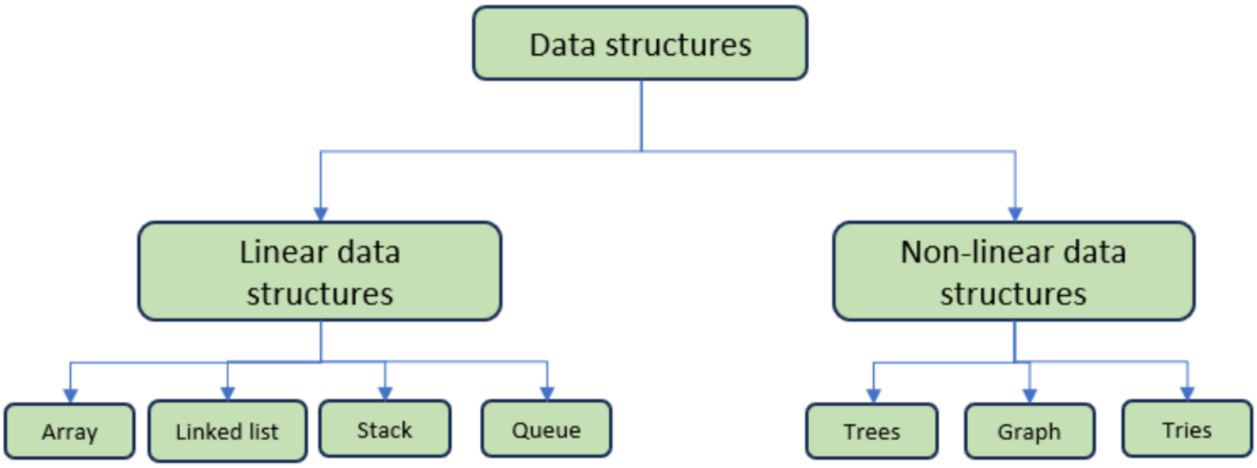
What is data?
Data definition defines a particular data with the following characteristics (Atomic, Traceable, Accurate, Clear and Concise).
Data type: Classify various types of data such as integer, string, etc. which determines the values that can be used with the corresponding type of data.

- **Built in data type**: Data types which a language has built-in support.(Int, Boolean (True, False), Floating, Character & string)
- **Derived data type**: Data types which are implementation independent as they can be implemented in one or the other way. (List, Array, Stack, Queue)

Algorithm: 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.
Not all procedures can be called as an algorithm. An algorithm should have following characteristics:

- **Unambiguous**: Clearly having the process & steps with inputs/outputs
- **Input**: An algorithm should have 0 or more well-defined inputs.
- **Output** – Algorithm should have 1 or more well-defined outputs and should match expected output.
- **Finiteness**: Algorithm must terminate after a finite number of steps.
- **Feasibility**: Should be feasible with available resources.
- **Independent**: Algorithm should have a step-by-step direction, which should be independent of any programming.

What are data structures?
Data structures is a systematic way to organize the data in order to use it efficiently. The data structures are language agnostic. It is a set of algorithms that we can use in any programming language to structure the data in memory.
Types of data structures:



1. **Linear data structures:**
    - The data stored in linear data structures sequentially. These are rudimentary structures since the elements are stored one after the other without any mathematical operations.
        - **Static data structures:**
            - In static linear data structures, the memory allocation is not scalable. Once the entire memory used, no more space can be retrieved to store more data.
            - Memory is required to be reserved based on the size of the program.
        - **Dynamic data structures:**
            - In dynamic linear data structures, the memory allocation can be done dynamically when required.
            - Advantages: Efficient considering the space complexity of the program.
    - Linear data structures are:
        - Arrays (Day 5 & 6)
        - Linked lists (Day 10 & 11)
        - Stacks (Day 13)
        - Queues (Day 16)
    - Advantages: Easy to implement
    - Disadvantages: Time & space complexity increases as the size of data increases.

2. **Nonlinear data structures**: Non-linear data structures store the data in the form of a hierarchy. Therefore, in contrast to the linear data structures, the data can be found in multiple levels.
        - Non-linear data structures are:
            - Graphs
            - Trees (Day 20)
            - Tries (Day 2)
            - Maps (Day 8)

Need for data structure:
As application and software's are getting complex and data intensive, three common problems are faced:

- **Data search**: Consider an inventory of 1 million (10^6) items of store, if the application is to search an item, it has to search an item in 1 million items every time slowing down the search. As data grows the search will become slower
- **Processor speed**: Processor speed although being very high, falls limited if the data grows to billion records.
- **Multiple requests** – As thousands of users can search data simultaneously, even the fast server fails while searching the data.

To solve the above-mentioned problems, data structures come to 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.

Execution time cases:
There are three cases which are usually used to compare various data structures execution time in a relative manner:

- **Worst case**: This is the scenario where a particular data structure operation takes maximum time it can take. If an operation's worst-case time is f(n) then this operation will take not take more than f(n) time in execution
- **Average case**: This is the scenario where the average execution of time of an operation of a data structure. If an operation takes f(n) time in execution, then m operations will take mf(n) time
- **Best case**: This the scenario where the least possible execution time of an operation of a data structure.

Fundamental operations 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 item
- **Update**: Updates an existing item
- **Delete**: Delete an existing item

Algorithm complexity analysis:

- **Time factor**: Time is measured by counting the number of key operations such as comparisons in the sorting algorithm.
- **Space factor**: Space is measured by counting the maximum memory space required by the algorithm.