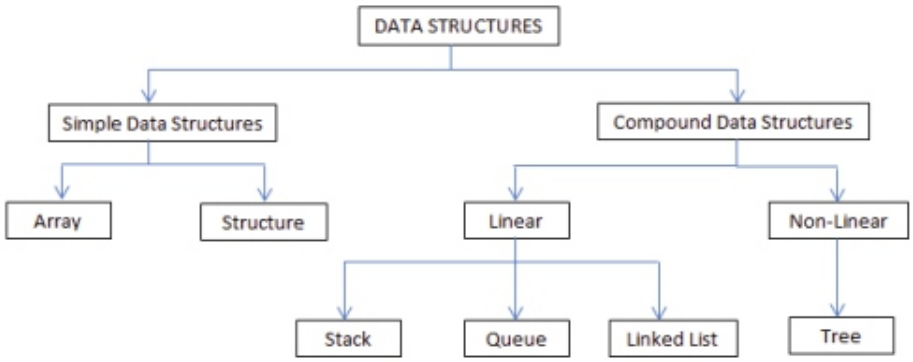
Chapter 1

Introduction to Data Structures

Data Structure is a way of collecting and organising data in such a way that we can perform operations on these data in an effective way. Data Structures is about rendering data elements in terms of some relationship, for better organization and storage.Data structure is a logical and mathematical view of any organisations data. For example, we have data player's name "Virat" and age 26. Here "Virat" is of String data type and 26 is of integer data type. We can organize this data as a record like Player record. Now we can collect and store player's records in a file or database as a data structure. For example: "Dhoni" 30, "Gambhir" 31, "Sehwag" 33

Classification of Data Structures:



Simple Data Structures:

These are normally built from primitive data types like integers, real, character, Boolean etc. There are following two types of simple data structures

1. Array
2. Structure

Compound Data Structures:

Simple data structures can be combined in various ways to form more complex structures called compound data structures.

They are classified into the following two types:

1. Linear data structures

These data structures are single level data structures. A data structure is said to be linear if its elements form a sequence. There are the following types:

a. Stack
b. Queue
c. Linked List

2. Non-linear data structures

These are multilevel data structures. Examples of non-linear data structure are Tree

and Graph.

Operations on Data Structures: The basic operations that are performed on data structures are as follows:
Insertion: Insertion means addition of a new data element in a data structure.
Deletion: Deletion means removal of a data element from a data structure if it is found.
Searching: Searching involves searching for the specified data element in a data structure.
Traversal: Traversal of a data structure means processing all the data elements present in it.
Sorting: Arranging data elements of a data structure in a specified order is called sorting.
Merging: Combining elements of two similar data structures to form a new data structure of the same type, is called merging.

Algorithm: An algorithm is a finite set of instructions or logic, written in order, to accomplish a certain predefined task. Algorithm is not the complete code or program, it is just the core logic (solution) of a problem, which can be expressed either as an informal high level description as pseudocode or using a flowchart.

An algorithm is said to be efficient and fast, if it takes less time to execute and consumes less memory space. The performance of an algorithm is measured on the basis of following properties:

Space Complexity
Time Complexity

Space Complexity

It's the amount of memory space required by the algorithm, during the course of its execution. Space complexity must be taken seriously for multi-user systems and in situations where limited memory is available. An algorithm generally requires space for following components:

Instruction Space: It's the space required to store the executable version of the program. This space is fixed, but varies depending upon the number of lines of code in the program.
Data Space: It's the space required to store all the constants and variables value.

Time Complexity

Time Complexity is a way to represent the amount of time needed by the program to run to completion.

Time Complexity of Algorithms:
Time complexity of an algorithm signifies the total time required by the program to run to completion. The time complexity of algorithms is most commonly expressed using the big O notation.

Time Complexity is most commonly estimated by counting the number of elementary functions performed by the algorithm. And since the algorithm's performance may vary with different types of input data, hence for an algorithm we usually use the worst-case time complexity of an algorithm because that is the maximum time taken for any input size.

Important Points

- Data structure is a logical and mathematical view of any organisations data.
- Simple data structures can be combined in various ways to form more complex structures called compound data structures.
- An algorithm is a finite set of instructions or logic, written in order, to accomplish a certain predefined task.
- Space complexity is the amount of memory space required by the algorithm, during the course of its execution. Space complexity must be taken seriously for multi-user systems and in situations where limited memory is available.

Exercise

Objective type questions.

Q1.When determining the efficiency of algorithm, the space factor is measured by

a. Counting the maximum memory needed by the algorithm
b. Counting the minimum memory needed by the algorithm
c. Counting the average memory needed by the algorithm
d. Counting the maximum disk space needed by the algorithm

Q2. For an algorithm the complexity of the average case is

a. Much more complicated to analyze than that of worst case
b. Much more simpler to analyze than that of worst case
c. Sometimes more complicated and some other times simpler than that of worst case
d. None or above

Q3. When determining the efficiency of algorithm the time factor is measured by

a. Counting microseconds
b. Counting the number of key operations
c. Counting the number of statements
d. Counting the kilobytes of algorithm

Q4. Which of the following data structure is linear data structure?

a. Trees
b. Graphs
c. Arrays
d. None of above

Q5. Which of the following data structure is non linear data structure?

a. Arrays
b. Linked lists
c. Both of above
d. None of above

Short answer type questions.

Q1. What is Data Structure ?
Q2. What are the two main measures for the efficiency of an algorithm ?
Q3. Why time complexity is important ?
Q4. Give example of Linear data structure.

Essay type questions.

Q1. How Space Complexity can be calculated ?
Q2. What are the uses of data structure?
Q3. Explain compound data structures ?

Answers

Ans1. a Ans2. c Ans3. b
Ans4. c Ans4. d