# Data Structures and Its Limitations

VISHAL V. MEHTRE [1], UDBHAV SINGH [2]
[1] *Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth Deemed University, College of Engineering, Pune, India*
[2] *Assistant Professor, Department of Electrical Engineering, Bharati Vidyapeeth Deemed University, College of Engineering, Pune, India*

***Abstract -- In this growing age of technology, computers play one of the most important roles. Be it medical surgeries to handling of huge power grids and power generation stations, all are controlled by computers. The most basic requirement of a computer to carry out its task is data. The data that is provided is initially in the raw form, but in order for it to work with other machinery that is required to carry out the task and produce an output the data needs to be structured. This structuring of data is done by some simple commands and algorithms termed as data structure. [1] This review paper aims at introducing different types of data structures as well as addressing the shortcomings of certain data structure. I will also try to point out certain improvements that can be carried out in certain data structures.***

## I. INTRODUCTION

DATA STRUCTURES:
Data structure is a method of structuring of data for easy usage and retrieval. The basic aim of data structure is to collect data values at a single place, establish relationship between the data and the different functions and operations that can be carried out on the collected data. [2]The various types of data structures are mentioned below along with a brief introduction to each:-

LINEAR DATA STRUCTURE:
1. Array: - An array is the most basic data structure. It stores values adjacent to each other ie.

Contiguous memory locations. The address of the data value next to the selected Data value can be easily retrieved by incrementing the address of the selected Data value by one. The data types of all elements in array are same.

Real life example: - Array can be understood similar to a staircase where the first Stair is the base value of the array and each and every subsequent data value is Similar to following stairs.

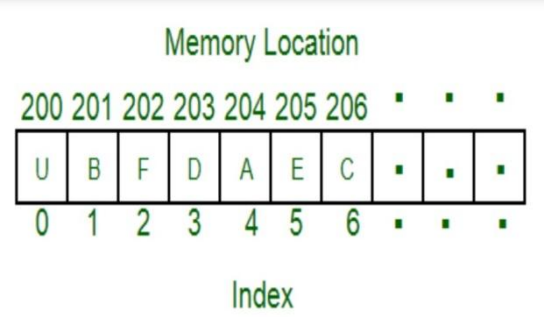


Fig1. Basic structure of array

2. Stack: - Stack is a type of linear data structure ie. All the operations that are to be performed on the data values are done In a particular order. The order that may be Followed LIFO (Last in First Out) or FILO (First in Last Out). [5] In stack the data values are added on top of each other and so there is a top value that keeps a track on the number of elements present in the stack. Real life example: - Books vertically placed on top of each other.

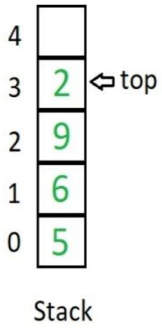


Fig2.Basic structure of stack

3. Queue: - Queue is another linear data structure. But unlike stack the order here is FIFO (First in First Out). In queue there are two parts front and rear, the rear is where the data values are added whereas the front is from where the data values are removed.

Real life example:-It's similar to people standing in a bank queue where people from the front are removed after depositing money and more people are added from the rear.

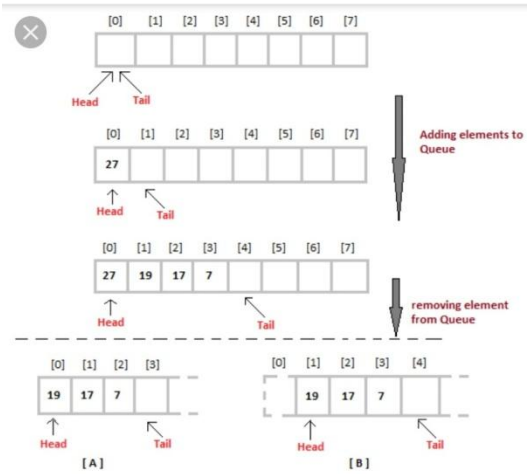


Fig 3.Basic structure of queue showing insertion and deletion operation

4. Linkedlist: - A linked list is another linear data structure but unlike other data structures Data is stored at noncontiguous memory locations. [4] Pointers are used for linking Different elements in a linked list. A linked list is made up of nodes where each And every node consists of two parts data field as well as reference link to the next Node.

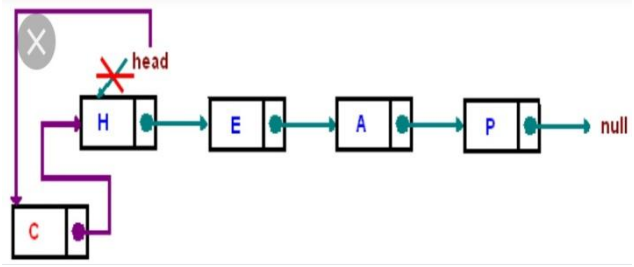


Fig 4.Basic structure of linked list

NON LINEAR DATA STRUCTURE

5. Graph: - A graph is an example of a non linear data structure. A graph is made up of Nodes(vertices) and edges(lines or arcs). The edges connect two or more nodes in a graph.

Real life example:-When we use UBER to take rides from one place to another, all the available routes are shown using graph data structure whereas the shortest Route is shown using the Dijkstra's algorithm. [6]

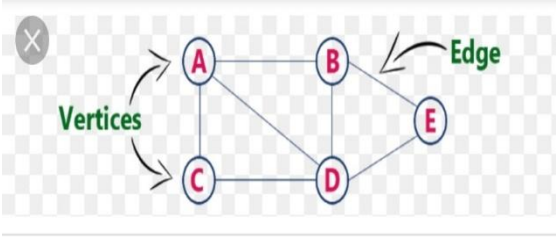


Fig 5. Basic structure of Graph

6. Tree: - A tree is a data structure which comes under ADT (Abstract Data Type). In tree there is a single node termed as parent node which stores the root value. Each Parent node can have multiple children (nodes). Each node is a data structure which

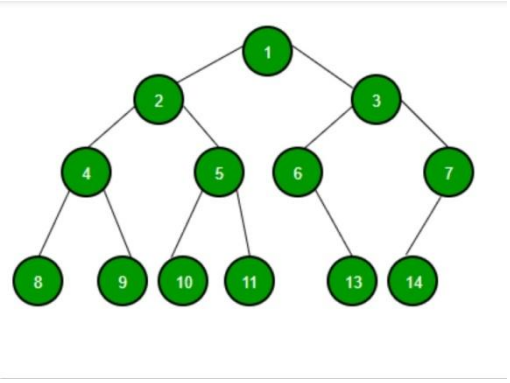


Fig 6. Basic structure of tree showing children nodes

Stores the data value and the reference to the next children nodes. The only Condition is that it does not have reference to the parent node. Each node acts as Parent node for its children and the tree continues.

There are different types of derived data structure from trees like heap, trie, binary Tree, binary search tree, n-ary tree, etc.

Advantages of data structure:-

1. Information can be stored on hard disks.
2. Allows management of database
3. Essential for algorithm design
4. Allows safe data storage
5. Allows easier data processing

Drawbacks of data structure:-

1. The algorithms are quite advanced.
2. If there is any problem in data structure it can be solved only by experts.
3. Data structures may work slowly with some data types.

Application of data structure:-

1. Data structures like linked list and Hash maps are used for memory management in OS design.
2. Binary trees are used in Database Design.
3. Trees are used in file system design [7]

## II. CONCLUSION

We will focus mainly on the third drawback as the other two drawbacks are quite relative with respect to the user.

The correct use of data structure depends on the requirement of the user. Correct use of data structure is a tradeoff between the efficiency and speed. Some data structures work well with one type of data whereas other data structures work well with other data type.

This is because different data structures use different methods and procedures for structuring data and this result in tradeoff between different properties of each data structure. [3]

For example an array is the simplest data type. They are fast, simple, and easy to search and sort but at the same time they are slow to expand, add or delete from as in order to carry out these functions we need to move all the elements of the array by one value.

On the other hand there are linked lists which are sortable and fast to search but they tend to slow down with large amounts of data.

Then there are trees that take time to setup but fastest if you have lots and lots of data.

The best example is a self-expanding data structure which can replace array such that we don't need to restructure the elements to make it more efficient ie. We need a data structure that can self-expand as well as structure its data according to given condition all by itself.

## ACKNOWLEDGEMENT

We would like to express our special thanks and gratitude to Dr. D. S. Bankar, Head, and Department of Electrical Engineering for their guidance and support for completing my research paper. I would also like to thank the faculty members of the department of electrical engineering who helped us with extended support.

## REFERENCES

[1] "Charles E. Leiserson, Clifford Stein, Ronald Rivest and Thomas Cormen" 'Introduction to algorithms'

[2] "Sumita Arora", 'Introduction to C++'

[3] "V.R. Kanagavalli, G. Maheeja", 'A study on the usage of data structure in information retrieval' by, arxiv, 2016

[4] "Zhang M., Zhao H., Wang T., Song G." 'Experimentation on data structure and algorithm', Higher Education Press, Beijing, 2011, Google Scholar

[5] "Victor Adamchik" 'Data structures and algorithm in Pen-based computing environment', Carnegie Mellon University, IEEE Global Engineering Education Conference (EDUCON), 2011

[6] "Jinhao Lu, Chi Dong", 'Research of shortest path algorithm based on data structure' IEEE International Conference, 2012

[7] "Richard F. Gillberg and Behrouz A. Frouzan", 'Data structures: A pseudocode approach'.