**Episteme: an online interdisciplinary, multidisciplinary & multi-cultural journal**

**Bharat College of Arts and Commerce, Badlapur, MMR, India**

**Volume 7, Issue 2** **September 2018**

# *DATA STRUCTURES: OVERVIEW AND ITS APPLICATIONS*

**By Deepika Ratnakar**

## Abstract

This paper tries to throw light in the types of data structures and their application in the field of Computer Science. Computer Science is an area of study which is gaining momentum as the need and urge for retrieving and exploring information is growing day by day. Data structures have been the area of research for a long period in the arena of computer science. The need to have efficient data structures has become even more important as the data grows in an exponential nature.

**Keywords**

Introduction, Types of data structure and their application

## Introduction

A **data structure** is a storage format and data organization that enables modification and efficient access.It provides a means to manage large amounts of data efficiently for uses such as internet indexing services and large databases. Usually, efficient data structures are key to designing efficient algorithms. Data structures are generally based on the ability of a computer to store and fetch data at any place in its memory, specified by a pointer—a bit string, representing a memory address, that can be itself stored in memory and manipulated by the program. Data structures can be used to organize the retrieval and storage of information stored in both main memory and secondary memory

## Types of data structure and their application

- Array
- Link List
- Stack
- Queue
- Map
- Tree
- Graph

### ➢ Array

Arrays are among the oldest and most important data structures, and are used by almost every program. They are also used to implement many other data structures, such as lists and strings. They effectively exploit the addressing logic of computers. In most modern computers and many external storage devices, the memory is a one-dimensional array of words, whose indices are their addresses. Processors, especially vector processors, are often optimized for array operations. Arrays are used to implement mathematical vectors and matrices, as well as other kinds of rectangular tables. Many databases, small and large, consist of (or include) one-dimensional arrays whose elements are records.

**Types of Array**

- **One-Dimensional Array**

It is used represent the elements of the array. The [] is used for dimensional or the sub-script of the array.

- **Two-Dimensional Array**

The Two -Dimensional array is used for representing the elements of the array in the form of the Columns and rows and these are used for representing the Matrix

- **Multi-Dimensional Array**

The Multidimensional Array are used for Representing the Total Number of tables of Matrix.

**Application of Array**

- **Array Can be Used in CPU Scheduling**

Arrays can be used in various CPU scheduling algorithms.

- **Array Can be Used in Recursive Function**

The function calls another function or the same function again then the current values are stores onto the stack and those values will be retrieve when control comes back.

- **Array are used to implement data structure**
  1.Stack using Array
  2.Queue using array

➢ **Link List**

A linked list is a dynamic data structure. The number of nodes in a list is not fixed and can grow and shrink on demand. Any application which has to deal with an unknown number of objects will need to use a linked list.

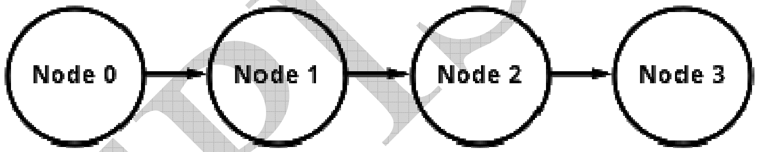
**Types of Linked Lists**

A **singly linked list**

In a **singly linked list**, each node in the **list** stores the contents and a pointer or reference to the next node in the **list**. It does not store any pointer or reference to the previous node. ... The last node in a **single linked list** points to nothing.

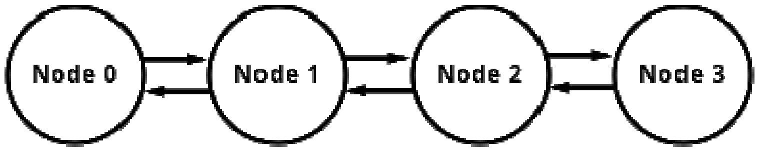
**Application of singly Link list**

1. In single linked list, element or nodes only link to the next element in the list. A single linked list is null terminating meaning when the pointer is null, the list is finished.



2. A **doubly linked list**

In a double linked list, each node contains a reference to the previous node and the next node (as long as they aren't the head or tail node.) Each node has a value property.



**Application of Linklist**

1. Consider the history section of web browsers, where it creates a linked list of web-pages visited, so that when you check history (traversal of a list) or press back button, the previous node's data is fetched.

2. Another real life example could a be queue/line of persons standing for food in mess, insertion is done at one end and deletion at other. And these operations happen frequent. dynamic queues / stacks are efficiently implemented using linked lists.

3.A simple real life example is a Train, here each coach is connected to its previous and next coach (Except first and last). In terms of programming consider coach body as node value and connectors as links to previous and next nodes.

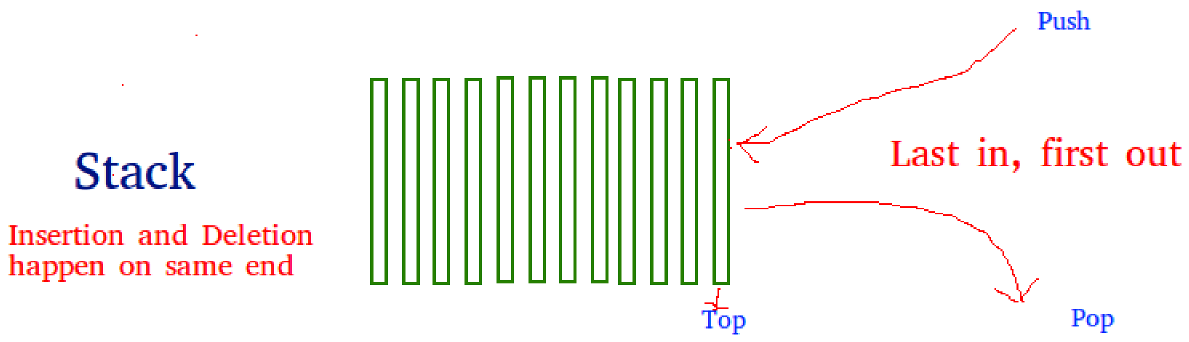
- **Stack**

It is a linear data structure which follows a particular order in which the operations are performed. The order may be LIFO(Last In First Out) or FILO(First In Last Out).

Mainly the following three basic operations are performed in the stack:

- **Push:** Adds an item in the stack. If the stack is full, then it is said to be an Overflow condition.
- **Pop:** Removes an item from the stack. The items are popped in the reversed order in which they are pushed. If the stack is empty, then it is said to be an Underflow condition.
- **Peek or Top:** It Returns top element of stack.

- **isEmpty: It** Returns true if stack is empty, else false.



**Application of stack data structure :**

- To **reverse a word**.
- An **"undo"** mechanism in text editors; this operation is accomplished by keeping all text changes in a stack.
  - Undo/Redo stacks in Excel or Word.
- **Language processing** :
  - space for parameters and local variables is created internally using a stack.
  - compiler's syntax check for matching braces is implemented by using stack.
- A **stack of plates/books** in a cupboard.
- A **garage that is only one car wide**. To remove the first car in we have to take out all the other cars in after it.
- Wearing/Removing **Bangles**.
- **Back/Forward** stacks on browsers.

## ➢ Queue

A Queue is a linear structure which follows a particular order in which the operations are performed. The order is First In First Out (FIFO). A good example of a queue is any queue of consumers for a resource where the consumer that came first is served first.

**Application of Queue Data structure**

1. When a resource is shared among multiple consumers. Examples include Disk scheduling, CpuScheduling.
2. Round Robin Scheduling
3. Job Scheduling
4. Keyboard Buffer

➢ **Maps**

A map is a data structure and its majorly used for fast searching or look up data. It stores data in the form of key and value pairs where every key is unique

Key and values could be almost of any data type. For this discussion lets look at only simple type keys. Generally you would have keys that are floats,int strings etc. A simple example would be storing first names as keys and values as ids in a map. Names obviously would be strings and id's would be ints.

| **Key** | **Value** |
|---|---|
| John | 1 |
| Peter | 2 |

**Application of Map data structure**

1. Dictionaries builds using a **Hash Table Data Structure**
2. It is used for caches, database indexing, fast data lookup

➢ **Tree**

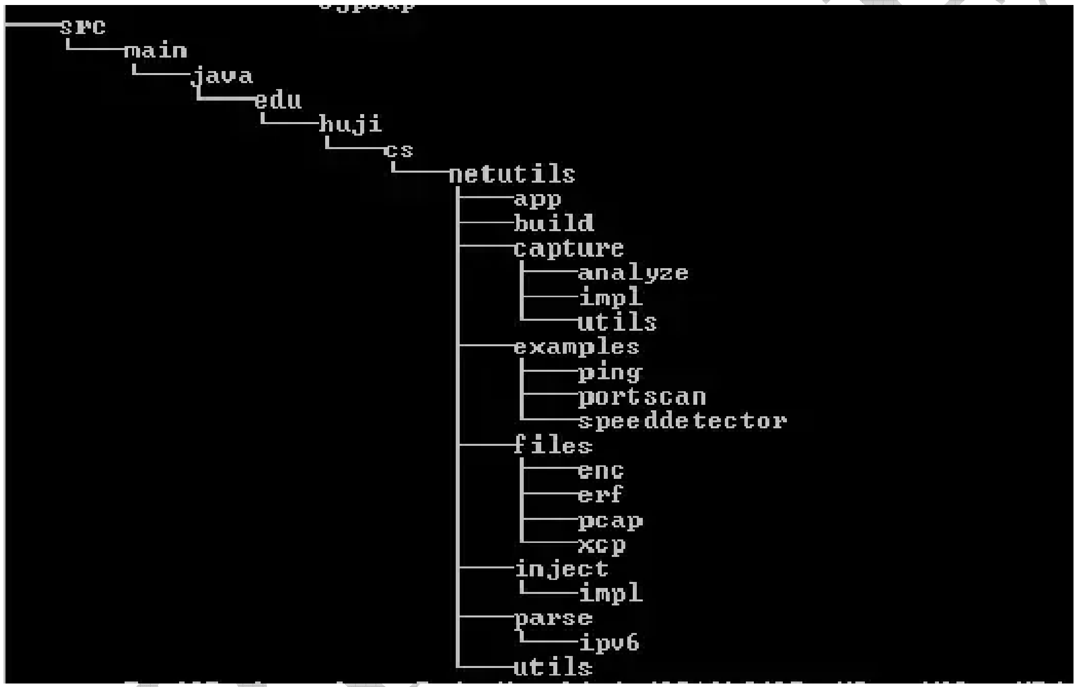
Tree represents the nodes connected by edges.

Binary Tree is a special datastructure used for data storage purposes. It has a special condition that each node can have a maximum of two children. It has the benefits of both an ordered array and a linked list as search is as quick as in a sorted array and deletion or insertion operation are as fast as in linked list.

**Application of Tree data structure**

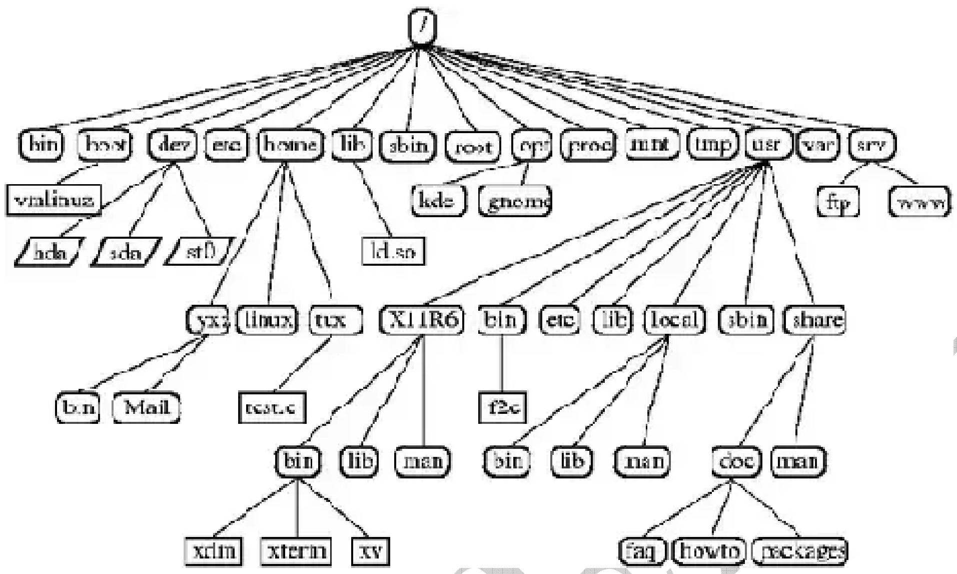
**1)Folders in Operating system**

In windows go to command line and type **tree.** You can see the folder tree structure.

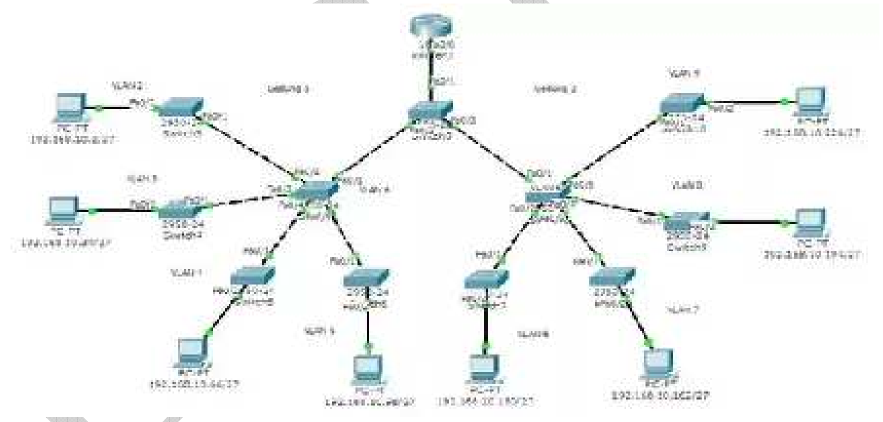


```
───src
   └───main
       └───java
           └───edu
               └───huji
                   └───cs
                       └───netutils
                           ├───app
                           ├───build
                           ├───capture
                           │   ├───analyze
                           │   ├───impl
                           │   └───utils
                           ├───examples
                           │   ├───ping
                           │   ├───portscan
                           │   └───speeddetector
                           ├───files
                           │   ├───enc
                           │   ├───erf
                           │   ├───pcap
                           │   └───xcp
                           ├───inject
                           │   └───impl
                           ├───parse
                           │   └───ipv6
                           └───utils
```

**2) Linux file system is also a tree**



**3) Network Routing**



➢ **Graph**

Graph is a pictorial representation of a set of objects where some pairs of objects are connected by links. The interconnected objects are represented by points termed as **vertices**, and the links that connect the vertices are called **edges**.

Formally, a graph is a pair of sets **(V, E)**, where **V** is the set of vertices and **E**is the set of edges, connecting the pairs of vertices

## Application of Graph Data structure

- Connecting with friends on social media, where each user is a vertex, and when users connect they create an edge.
- Using GPS/Yahoo Maps, to find a route based on shortest route.

## Conclusion

In this paper I have included types of **Data Structure** and**their application** .
Data Structure is to keep the **data** in-memory. So that it can be Queried, Fetched, Updated in a fast manner to suite the **application** needs Like Hash Table - used for fast **data** lookup - symbol table for compilers, database indexing, Caches,Unique **data** representation.

## References

- https://www.geeksforgeeks.org/
- https://www.quora.com

## Bio

**Deepika Ratnakar** is working in Bharat College as a Asst.Professor and also Visiting Faculty in B.K.Birla College.