

Asymptotic Analysis

Unit - 1.

Data Structure.

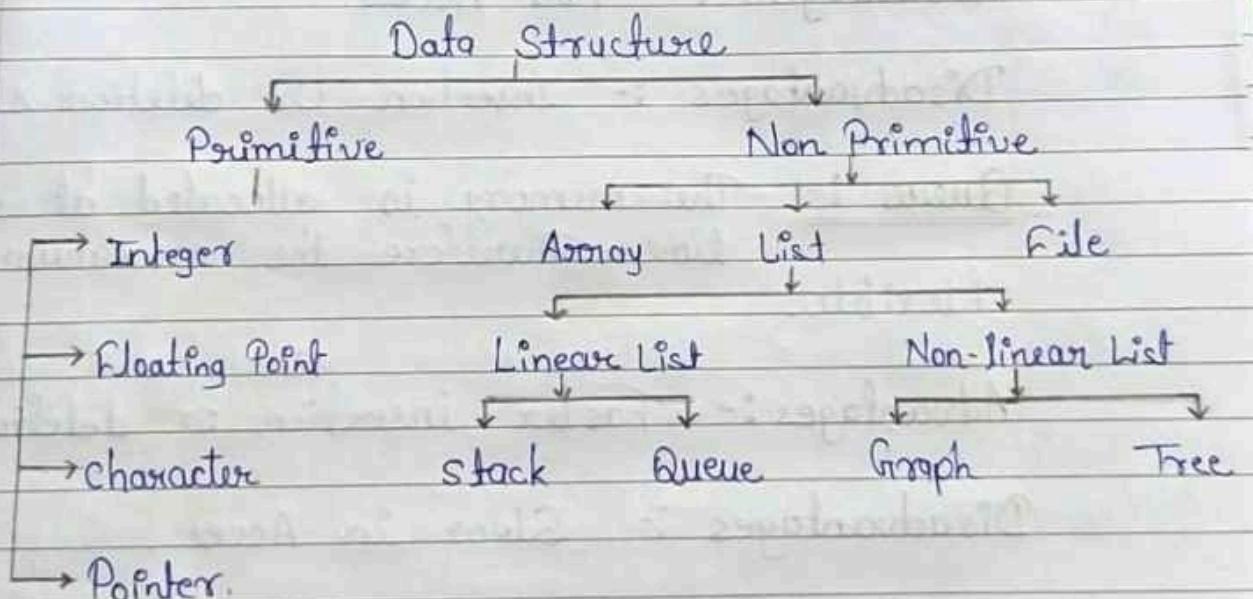
Data :- The quantities characters or symbols on which operations are performed by a computer which may be stored and transmitted in the form of electrical signals and recorded on magnetic, optical or mechanical recording medium.

Information :- The meaningful or process data is called information.

Definition of Data Structure :-

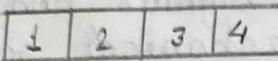
A data structure is the systematic way to organise data so that it can be used efficiently.

Classification of Data Structure -

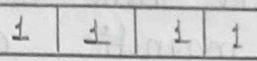


Classification of data Structure.

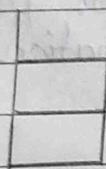
Linear Data Structure :- In linear data structure data elements are arranged sequentially or linearly where each element is attached to its previous or next elements.



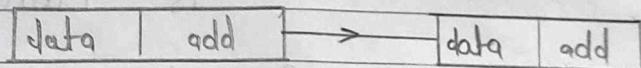
Array



queue



stack



Linked list.

Static :- The memory is allocated at compile time.

Advantages :- Fast Access.

Disadvantages :- Insertion Or deletion slow.

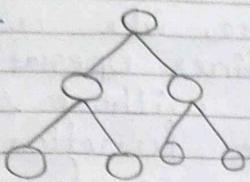
Queue :- The memory is allocated at run time. Therefore the maximum size is flexible.

Advantages :- Faster insertion or deletion.

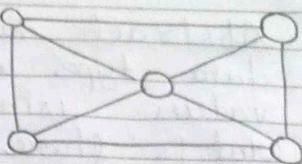
Disadvantages :- Slow in Access.

Non-Linear :- A data structure is non-linear then all the elements are not arranged in a linear or sequential order.

Ex:-



Tree.



Graph.

Advantage of Data structure

- (i) Efficiency :- Proper choice of data structure make program efficient in terms of Space and time.
- (ii) Reuseability :- One implementation can be used by multiple line program.
- (iii) Abstraction :- Data Structure is specified by an abstract data type which provide a level of abstraction it mean the client program does not worry about the implementation details.

Abstract data type (ADT)

User defined data type :-

The Operation and values of user defined data are not specified in the language it but it is specified by the users.

Ex. - structure & Union

Abstract data type are like user defined data type which defines operations on values using function without specifying what is there inside the function and how the operations are performed.

Ex.- Stack ADT (Abstract data type).

In other words, ADT is a black box which hide the inner structure and design of the data types from the users.

Operation of DS (Data Structure).

- ① Insertion
- ② Deletion
- ③ Traverse (visit)
- ④ Search
- ⑤ Sorting
- ⑥ Merging.

- ① Insertion - Insertion means adding new details or new node into the data structure.
- ② Deletion - Deletion means removing a node from the data structure.
- ③ Traverse - Traverse means accessing each node exactly once so that the data structure can be processed.

④ Searching - Searching means finding the location of nodes for given by value.

⑤ Sorting - Sorting means arranging the data in a particular Order.

⑥ Merging - Merging means union joining the two list.

Array :- Array are defined as the collection of similar types of data items stored at contiguous memory allocation. It is one of simplest data structure where each data elements can randomly accessed by using its Index number.

Array Properties :-

- Each element in array is same data type.
- Element in the array are stored at contiguous memory allocation from which first element is stored at the smallest memory allocation.
- Element of the array can be randomly accessed since we calculated the address of each elements of the array with the given address and the size of data elements.
- Index start with 0.
- The array length is $n-1$.
- Each element in the array can be accessed by it's index.
- Array are useful because sorting & searching

a value in an array is easier.

- Array are best to process multiple values quickly & easily.
- An array are good for storing multiple values in a single variable.

Some Operation are associated with array :-

- Creation of an array.
- Insertion of new element.
- Deletion of new element.
- Traversing of an array.
- Modification of an element.
- Merging of Array.

Classification / Representation of an Array :-

Array

One Dimensional
Array

Two Dimen-
sional
Array

Multi Dimen-
sional
Array

(i) One Dimensional Array :- A simple data structure that stores a collection of similar type data in a contiguous block of memory, also called single dimensional array.

- Stores data as a list.
- Syntax : data type []. name = new data type [size]

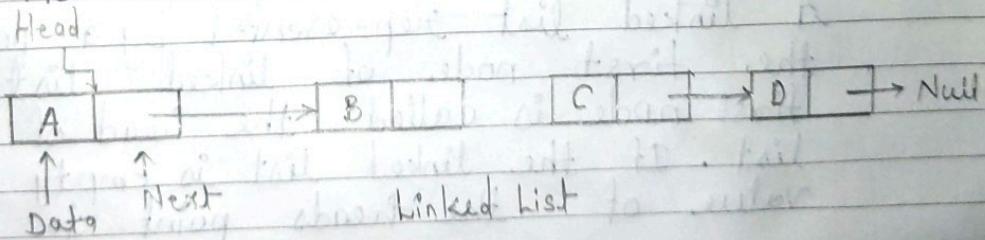
(ii) Two dimensional Array :- A type of array that stores multiple data elements of the same type in matrix or table like format with a number of rows and columns. called as Two dimensional arrays.

- Syntax : `data-type [][] name = new data-type [rows][columns];`
- Stores data in row-column format.

(iii) Multi dimensional Array :- Multidimension array is the array with two or more dimensions.

- Mostly two dimensional array are used in multi dimensional array.

★
Linked List :- Like array Linked list is a linear data structure. Unlike arrays. linked list elements are not stored at a contiguous location. the elements are linked using pointers. They include a series of connected nodes. Here each node stores the data the address of the next node.



-:- Advantages of linked list

- Dynamic Array.
- Easy of Insertion / Deletion.

-:- Drawbacks of linked list :-

- Random access is not allowed. We have to access elements sequentially starting from the first node (head node). So we cannot do a binary search with linked lists efficiently with its default implementation.
- Extra memory space for a pointer is required with each element of the list.
- Not cache friendly, since array elements are contiguous locations, there is Locality of reference which is not there in case of linked list.

:- Types of linked list :-

Linked List

Simple linked list Double linked list Circular linked list

Representation of linked list :-

A linked list represented by a pointer to the first node of linked list. The first node is called the head of the linked list. If the linked list is empty then the value of the heads point to Null.

- Each node in a list consist of atleast two parts :-

- 1o) A data Item. (We can store integer, string or any type of data).
- 2o) Pointer (or Reference) to the next node (connects one node to other) or an address of another node.

