Week #3

20 - 6 - 2022

Why we need data Structures?

We make data structures to make it easy to use and easy to fetch data of same type.

Array

→ Data ~~ty~~ structure of contigeous memory

Advantages

1. Easy to use (~~Insertion~~ Sort, Updation and Search)
2. Very fast access to data
3. Easily search and sort

Disadvantages

1. Cannot resize (Fixed size)
2. Memory wastage because of fixed size
3. Deletetion of index/node is very difficult
4. Cannot add new node in array
5. Same data type
6. Not very adaptable
7. Resizing an array is difficult (Not in vectors)
9. Insertion is difficult in empty spaces after deletion

Next — The next pointer inside a Node is link

Linked List (Collection of nodes

Advantage over array

1. Can be resize.
2. Insertion and deletion is easy.
3. Can be different data type.
4. Do not waste memory.
5. Size is not specfied at start
6. Data is accessed (or required) by head pointer

→ Self referential class

// Class which has its own pointer

Eg Node class

```
class Node { int data;
             Node * next; }
```

Memory management

Memmory managment in C++ is programmers responsibility. If programmer creates a point and reserve a chunk of memory in heap then after the execution of program. It is important to relise all the reserve memory

In case of Java, memory managment is IDEs responsibility.

Moving from one node to another by following a next reference is known as link hopping.

26-9-2022

Why array's index starts with '0'?

→ To locate the index and se the distance of index from start

Address of Array's index

Base address + (index * size of dataType)

Singly LinkedList Disadvantages

→ Can traverse only in forward direction

→ Cannot move backword.

→ Very time taking / more iteration to get required in deleting last node. ~~itra itration~~ to tail.

Doubly Linkedlist advantages over SLL

→ can move forward + backwarcl.

→ Insertion and deletion is easy.

→ Easily reverse display list.

→ Easily insert at any place in List.

→ No need to travers through the whole list (if sentinal tail is created) ~~Fand~~ needs to delete from tail.

Stack

A stack is a container of objects that are inserted and removed according to the last-in first-out (LIFO) principle. Objects can be inserted into stack at any time only at one side.