Academic Year: 2020

Semester 3

Course Code: CS-212L

Course Title: Data Structures and Algorithm Lab

CS-212L

Type of Lab: Open Ended **Weightage: 10%**

CLO 1: CLO's.

State the Rubric	**Cognitive/Understanding**	CLO1	Rubric A

Rubric A: Cognitive Domain

Evaluation Method: GA shall evaluate the students for Question according to the following rubrics.

CLO	**0**	**1**	**2**	**3**	**4**
CLO1					

Lab 1

Processing Steps:

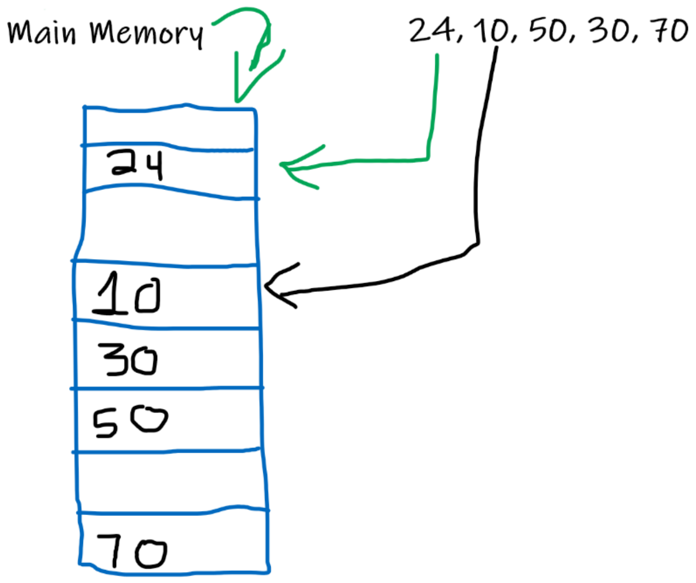
Step1: What is Node?

Node is nothing, but it is a fancy term of the following phrase “pointer reference to each element of given input values placed in somewhere in memory” i.e., Node represent each single value from an input set in main memory. It can also store more information along with the data (which we get from the input) like, it can have a pointer variable which points to the next node element of the same input data present in somewhere in memory. It can also store other information according to the requirement. Let’s picture it.

Suppose, we have the following input of data (data of a player who scores points in multiple games)

Player’s Points = [24, 10, 50, 30, 70]

Now, we want to use this data in our link list (you will learn it later on in this document), so we have 5 points of a player, so as we know Node is pointer reference to each element of a given set in the present in memory along with additional information.



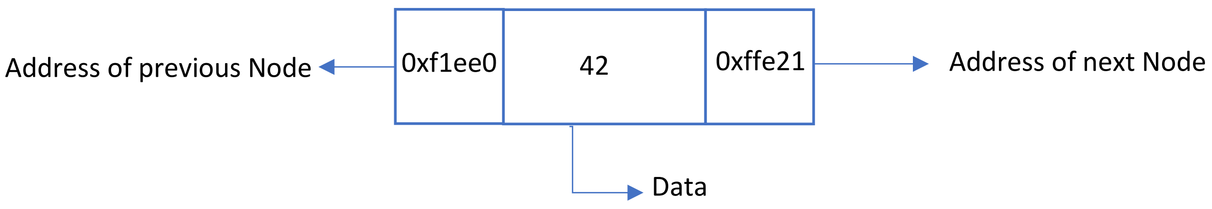
Here you can see each element is present in random positions in memory and each block represents a node.

A node in Link List:

This node has usually consisted of two parts,

- Data
- Next Pointer variable of Node type
- Previous Pointer variable of Node t

Pointer variable of Node type is used to store the next node from the given input. At the time of creation, it is pointing to *null.*



Step 2: Linked List:

Linked List is a linear, dynamic data structure, which came into existence to get rid of array Data structure, Insertion in arrays are not dynamic (or difficult to insert) also deletion in arrays requires a lot of effort. Arrays are static data structure which stores the particular information in a contiguous memory location.

Doubly Linked List:

It is extension on linked list with some extra feature, it had the previous pointer which points to the previous node.

Step 3: Why Doubly Linked list?

Linked List advantages over an array:

The linked list data structure is used to handle the above-mentioned problems which are present in an array. Linked List is dynamic in nature. Insertion, deletion is efficient rather than a regular Array data structure.

Doubly Linked list over comes some limitation of linked list,

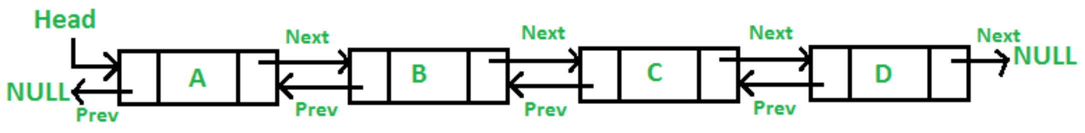
- it gives the back/reverse traversal easy
- it makes the delete function code efficient

Step 4: How Doubly Linked list implemented?

Linked List works in such a way that

- Takes each data from the input set
- Create a node of that data
- Finally, set the pointer variable according to the situation

As the elements are placed in random positions, and they are accessed or manipulated by using their addresses, so we can easily insert new data at any time just at the cost of attaching the address of that newly created node to the last created node and that's it.



Deletion is also very efficient as compared to arrays we can delete or unlink a specific node from our list by just setting the specific address of that node to null. And then free the space of that unlinked node from memory.

Step 5: Operations on Doubly Linked List:

- Insertion (At Head, At Tail, between two nodes)
- Deleting a node
- Searching a node

Step 6: Insertion:

In this operation we build the linked list, each data from the given set is taken and we create a node then linked to list (or create a new list). First, we will see the Insertion of a node in a linked list at the head

```
struct Node {

int data;
Node * next;
Node * previous;
}
```

Step 1: Create a new node of the given data

Step 2: Check List is empty, If empty then set "Head Node" value to the node you created earlier

Step 3: If List is not empty (Head Node has an address) then, do the following steps in-order

a) Set the 'next pointer variable' of the node created at step 1 = 'Head' value (address)
b) Set the 'Head' previous pointer variable = node created at step 1
c) Set the 'Root Node' value to the Node created at step 1

Step 7: Search:

This operation takes the input data and returns the node address which contains the data if not then it returns null

Step1: Check the Head Node data with the given data if matched then return the Head node address

Step2: If not match then create a temporary variable and assign the value of Head Node 'Next pointer variable'. Now temporary variable holds the address of the next node after Head node

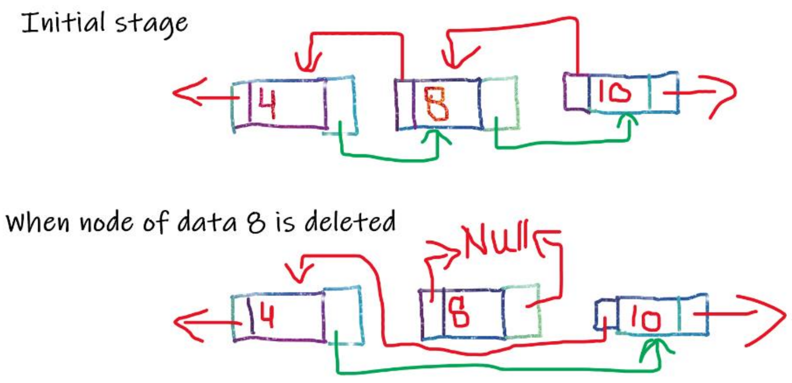
Step2: Run a loop which terminates when the temporary node 'Data' is equal to the given data or temporary node value is null

Step3: In-Loop body assign the temporary value of the temporary node 'next pointer', and check the new temporary value is null, if yes, then exit from the loop and your search is not found, if not the continue to the loop

Step4: If the loop terminates and temporary variable hold address instead of 'null'. Simply return it which is your desired node.

Lab Problem:

- Implement the Delete function. Hint [take help from the Search function and do some pointer shifting to delete/unlink the node from list].



- You will be given a data (which may or may not be present in the current state of Linked List), Find the specific position of that node if data is present, then ask the user to enter the value you want to update, if not then insert at the tail.

Assignment:

"ABC" is flower data website repository; this website has number of images and they want to provide user friendly view to the user. They had divided the images on multiple pages and they want to keep track of the first page to have information of its consecutive page and next page will also have the information of its next page as well as its previous page. The Last page have the information of first page

You are given a number of pages and each page have some information (pageName, numberOfImages). Write a code which take this input and the environment where

- first page node has connectivity to the next page and same for others,
- Your program should be able to delete a page at any time when user want
- Your program should be able to update the page information when user want
- Your program should be able to insert a new page either as a first page or somewhere between current pages or as a last page
- Your program should be able to display the pages and has the functionality to get pageName of given pageNumber
- Code should be properly commented, clean, free from exception errors and normal termination