SVKM's NMIMS
School of Technology Management & Engineering
Computer Engineering Department
Program: B.Tech\MBA Tech Sem II

Course: Data Structure and Algorithm
2021-2022

LAB Manual

PART A

(PART A : TO BE REFFERED BY STUDENTS)

Experiment No.04

A.1 Aim:

To study and implement concept of Linked list data structure.

A.2 Prerequisite:

1. Knowledge of different operations performed on linked list data structure.
2. Knowledge of different types of linked list and their applications.
3. Fundamental concepts of C\C++.

A.3 Outcome:

After successful completion of this experiment students will be able to

1. Identify the need of appropriate selection of data structure
2. Identify the steps of linked list data structure selection.
3. Implement different type of Linked list data structure to solve the given problem
4. Enlist the applications of Linked list data structure.
5. Differentiate between different types of Linked list.

A.4 Theory:

A.4.1. Introduction to Linked List

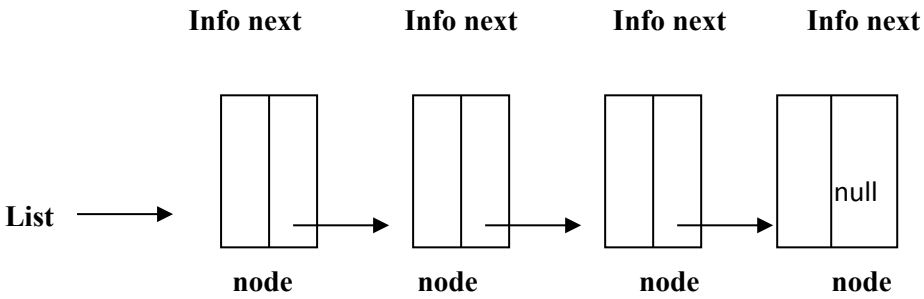
A.5 Procedure/Algorithm:

A.5.1:

What are the drawbacks of using sequential storage to represent stacks and queues? One major drawback is that a fixed amount of storage remains allocated to the stack or queue even when the structure is actually using a smaller amount or possibly no storage at all. Further no more than that fixed amount of storage may be allocated, thus introducing the possibility of overflow.

If the memory is allocated for the variable during the compilation (i.e. before execution) of a program, then it is fixed and cannot be changed. For example, an array A [100] is declared with 100 elements, then the allocated memory is fixed and cannot decrease or increase the SIZE of the array if required. So we have to adopt alternative strategy to allocate memory only when it is required. There is a special data structure called linked list that provides a more flexible storage system and it does not require the use of arrays.

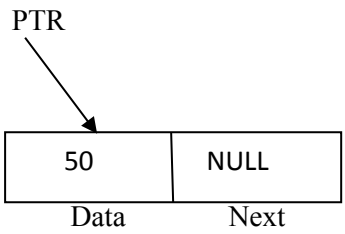
In a sequential representation, the items of a stack or queue are implicitly ordered by the sequential order of storage. Thus, if q.items[x] represents an element of a queue, the next element will be q.items[x+1]. Suppose that the items of a stack or q queue were explicitly ordered, that is, each item contained within itself the address of the next item, such an explicit ordering gives rise to a data structure as below, which is known as **Linear Linked List.**



Each item in the list is called a node and contains two fields, an ***information*** field and a ***next address*** field. The information field holds the actual element on the list. The next address field contains the address of the next node in the list. Such an address, which is used to access a particular node, is known as a pointer. The entire linked list is accessed from an external pointer ***list that*** points to the first node in the list. The next address field of the last node in the list contains a special value, known as ***null***, which is not a valid address. This ***null pointer*** is used to signal the end of the list.

The list with no nodes on it is called ***empty list*** or the ***null list***. The value to the external pointer list to such list is the null pointer. A list can be initialized to the empty list by the operation list=null.

Below figure shows a typical example of node.



PTR->Data=50

PTR->next=NULL

Advantages and Disadvantages

Linked list have many advantages and some of them are:

1. Linked list are dynamic structure. That is they can grow or shrink during the execution of a program.
2. Efficient memory utilization: In linked list representation, memory is not pre-allocated. Memory is allocated whenever it is required. And it is de-allocated when it is not needed.
3. Insertion and deletion are easier and efficient. Linked list provides flexibility in inserting a data item at a specified position and deletion of a data item from the given position.
4. Many complex applications can be easily carried out with the linked list.

Linked list has following disadvantages:

1. More memory: to store an integer number, a node with integer data and address field is allocated. That is more memory space is needed.
2. Access to arbitrary data item is little bit cumbersome and also time consuming.

Operations on Linked List

The primitive operations performed on the linked list are as follows:

1. Creation
2. Insertion
3. Deletion
4. Traversing
5. Searching
6. Concatenation

Types of a Linked list:

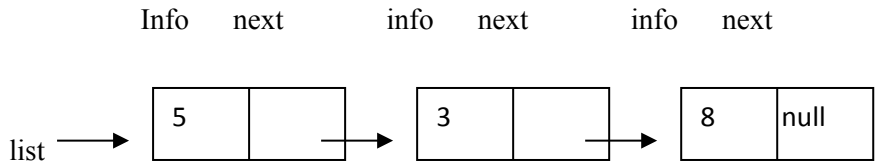
Basically we can divide the linked list into the following three types in the order in which they are arranged.

1. Singly linked list
2. Doubly linked list
3. Circular linked list

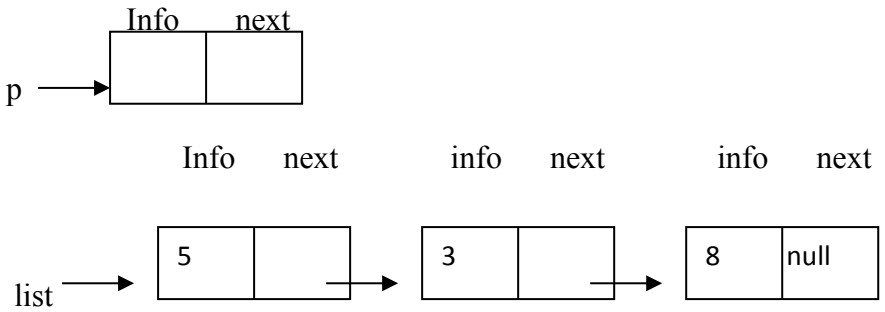
Inserting and removing nodes from a list:

1. A list is a dynamic structure. The number of nodes on a list may vary dramatically as elements are inserted and removed.

2. Suppose we are given a list of integers, as below and we desire to add the integer 6 to the front of the list.



3. The first step is to obtain a node in which to house the additional integer.
4. If a list is to grow and shrink, there must be some way for obtaining empty nodes to be added onto the list.
5. Unlike an array, a list does not come with pre-supplied set of storage locations into which elements can be placed.
6. Let us assume the existence of a mechanism for obtaining empty nodes. The operation p=getnode(); obtains an empty node and sets the contents of a variable named p to the address of that node.
7. The value of p is then a pointer to this newly allocated node.



8. The next step is to insert the integer 6 into the info portion of the newly allocated node. This is done by the operation info(p)=6;

TASK 1:

Q1. **Write a C/C++ program to implement the following operations on a Singly linked list:**

1. Create a linked list (Beginning, end, middle)
2. Insert a new node (Beginning, end, middle)
3. Modify the info of a node (position to be specified by the user)
4. Delete a node (position to be specified by the user)
5. Traverse the list

PART B

(PART B : TO BE COMPLETED BY STUDENTS)

(Students must submit the soft copy as per following segments within two hours of the practical. The soft copy must be uploaded on the Blackboard or emailed to the concerned lab in charge faculties at the end of the practical in case the there is no Black board access available)

Roll No.	Name:
Class :	Batch :
Date of Experiment:	Date of Submission
Grade :	Time of Submission:
Date of Grading:	

B.1 Software Code written by student:

(Paste your code completed during the 2 hours of practical in the lab here)

B.2 Input and Output:

(Paste your program input and output in following format, If there is error then paste the specific error in the output part. In case of error with due permission of the faculty extension can be given to submit the error free code with output in due course of time. Students will be graded accordingly.)

B.3 Observations and learning [w.r.t. all tasks]:

(Students are expected to comment on the output obtained with clear observations and learning for each task/ sub part assigned)

B.4 Conclusion:

(Students must write the conclusion as per the attainment of individual outcome listed above and learning/observation noted in section B.3)

B.5 Question of Curiosity

(To be answered by student based on the practical performed and learning/observations)

Q1. State the advantages of linked list.