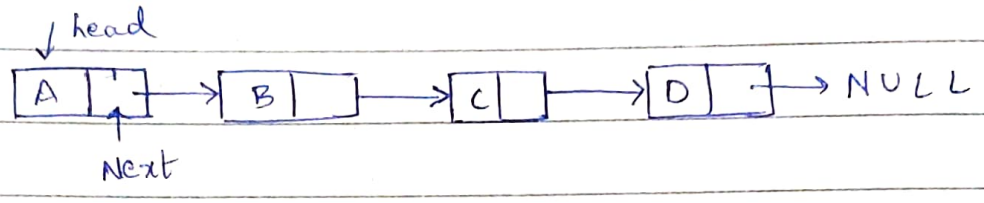
Date.....
Page.....

DSA LAB-5

Theory

1)A. A linked list is a linear data structure, in which elements are stored at contiguous memory locations. The elements in a linked list are linked using pointers as shown in below image.



In simple words, a linked list consists of nodes where each node contains a data field and a reference (link) to the next node in the list.

Features of linked list over array:

1) we can add elements in the list if necessary but whereas in array it is not possible.
2) Linked list provides an efficient way of storing related data and perform basic operations such as insertion, deletion and updating of information. But this is not at all possible in case of arrays.

Date .....
Page .....

2) A. To implement a single linked list, we basically need three pointers.

1) A 'head' pointer which is used for pointing to the start of the record in a list.
2) A 'tail' pointer which is used for pointing last node is that its subsequent pointer points to nothing (NULL).
3) A pointer in every node is used for pointing to the next node element.

3) A. There are four types of linked list. They are

1) ~~Link~~ Singly Linked list
2) Doubly linked list
3) Circular linked list
4) Doubly Circular linked list

1) Singly linked list:

It is simplest type of linked list in which every node contains some data and a pointer to the next node of same data type. The node contains a pointer to the next node means that node stores the address of the next node in the sequence.

A single linked list allows traversal of data in only one way (forward).

2) Doubly linked list:

A Doubly linked list or a two-way linked list is a more complex type of linked list which contains a pointer to the next as well as previous node in sequence. Therefore, it contains three parts of data, a pointer to the next node, and a pointer to the previous node.

A Doubly linked list allows traversal of data in both directions (both forward and backward).

3) Circular linked list:

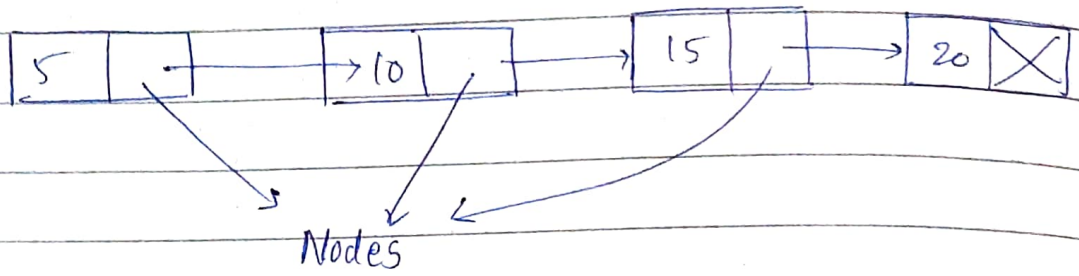
A circular linked list is that in which the last node contains the pointer to the first node of the list. While traversing circular list, we can begin at any node and traverse the list in any direction forward and backward until we reach the same node we started.

A circular linked list has no beginning and no end.

4) Doubly Circular linked list:

A Doubly Circular linked list or a circular two-way linked list is more complex type of linked list that contains a pointer to the next as well as previous node in sequence. The difference between the doubly linked and circular doubly linked list is the same as that between a singly linked and circular linked list.

Linked list nodes are used to point next node by means of points. So the nodes of linked list are represented using a pointer.



4) A.

Singly Linked List	Doubly Linked List
1) Singly linked list has only a data field and next link field.	1) Doubly linked list has a data field and two field previous link field and next link field.
2) In singly linked list traversal can be done using the next node only.	2) In doubly linked list, traversal can be done using both previous and next nodes.
3) SLL occupies less memory than DLL as it has 2 fields	3) DLL occupies more memory as it has 3 fields.
4) Less efficient access to elements.	4) More efficient access to elements.

5) A. Applications of linked lists:

1) Implementation of stacks and queues.
2) Implementation of graphs.
3) Dynamic memory allocation.
4) Maintaing directory of names.
5) Performing arithmetic operations on long integers
6) Manipulation of polynomials by storing constants in the node of linked list.
7) representing sparse matrices.

6) A. Algorithm

1) for counting number of employees.

Step 1: Start
step 2: Set count = 0. Intialize variable count.
Step 3: Set Ptr = Start. (Intialize)
step 4: Repeat following steps while Ptr != NULL
    4.1: Set count = count + 1
    4.2: Set ptr = ptr → next
step 5: write count
step 6: End

Date
Page

2) for transvering through linked list:

Step 1: Start
Step 2: Set ptr = Start [Initialize]
Step 3: Repeat steps (following) while ptr != NULL
    3.1: Apply process to ptr → DATA
    3.2: Set ptr = ptr → next
Step 4: End