

Date : 7th - 10 - 2020

Morning Session : 9am – 11.00 PM

By ~ Rohan Kumar

Topics: Linked List -2

Linked List Operations: Traverse, Insert and Delete

Two important points to remember:

- head points to the first node of the linked list
- next pointer of the last node is NULL, so if the next current node is NULL, we have reached the end of the linked list.

How to Traverse a Linked List

Displaying the contents of a linked list is very simple. We keep moving the temp node to the next one and display its contents.

When temp is NULL, we know that we have reached the end of the linked list so we get out of the while loop.

How to Add Elements to a Linked List

You can add elements to either the beginning, middle or end of the linked list.

Add to the beginning

- Allocate memory for new node
- Store data
- Change next of new node to point to head
- Change head to point to recently created node

Add to the End

- Allocate memory for new node
- Store data
- Traverse to last node
- Change next of last node to recently created node

Add to the Middle

- Allocate memory and store data for new node
- Traverse to node just before the required position of new node
- Change next pointers to include new node in between

How to Delete from a Linked List

You can delete either from the beginning, end or from a particular position.

Delete from beginning

- Point head to the second node

Delete from end

- Traverse to second last element
- Change its next pointer to null

Delete from middle

- Traverse to element before the element to be deleted
- Change next pointers to exclude the node from the chain

To see linked list implementation in please watch Recorded Lecture

[Recorded Lecture](#)

Instructor Provided Link:

<https://www.geeksforgeeks.org/doubly-linked-list/>

MCQ 1

Time Complexity ->traversing linked list

- a. $O(n)$
- b. $O(1)$
- c. $O(\log n)$
- d. $O(n*n)$

Answer: A, $O(n)$

MCQ 2:

Time Complexity -> searching an element in linked list

- a. $O(n)$
- b. $O(1)$
- c. $O(\log n)$
- d. $O(n*n)$

Answer: A, $O(n)$