



Your Ultimate Guide To Landing
Top AI roles



Introduction to Linked List

→ A Linked List is a linear data structure where elements (nodes) are stored in separate memory locations and are connected to each other.

Example: Store a list of numbers [5, 10, 15, 20] in Linked List



→ Difference between Arrays and Linked List?



→ Linked List → Collection of **Nodes**



→ Node contains **data** and pointer to the next node.

→ In Linked List, nodes can contain

① Homogeneous data

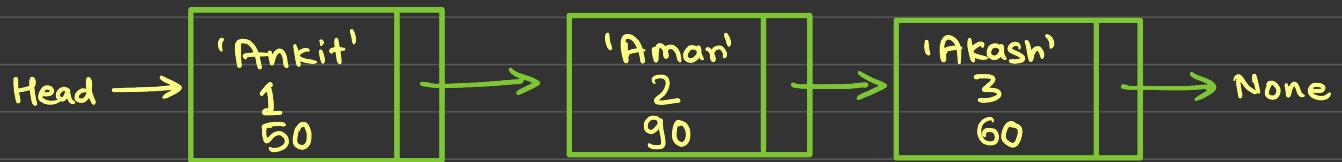


② Heterogeneous data



③ Record Data

→ Student Record data → Student (Name, Rollno, Marks)



→ Linked Lists has following properties

- ① Successive nodes are connected by pointers
- ② Last element points to None
- ③ It can be made just as long as required
- ④ Head points to the first node.

→ Operations on a Linked List

Operation	Time Complexity
Insert at beginning	$O(1)$
Insert at Position (ith)	$O(n)$
Delete at beginning	$O(1)$
Delete at Position (itm)	$O(n)$
Search node	$O(n)$
Display all items	$O(n)$

→ Insertion in Linked List

Head → None

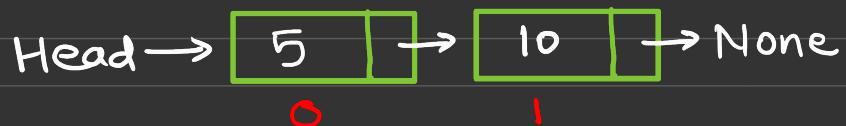
① Insert(10)

- Create an empty node (temp)
- Add value(10) and next(None)
- Make head point to temp



② Insert(5, 0) ← Insert in the beginning.

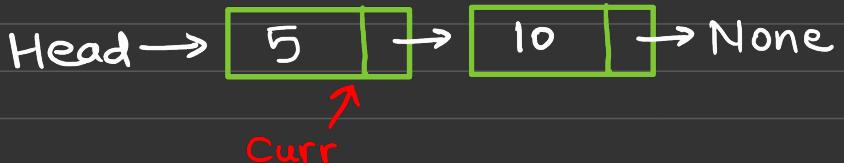
- Create an empty node (temp)
- Add value(5) and next(Head)
- Make head point to temp



③ Insert(2, 1) ← Insert in between

→ Init curr = Head.

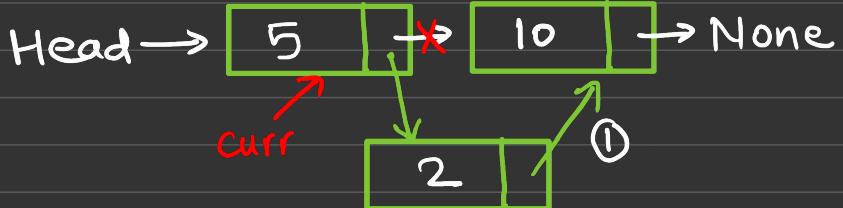
→ Iterate curr such that it points (i-1)th node.



→ Create an empty node (temp)

→ Add value(2) and next(curr.next)

→ Make Curr.next point to the temp.



④ Insert(6,1) ← Insert in the end

→ Init curr = Head.

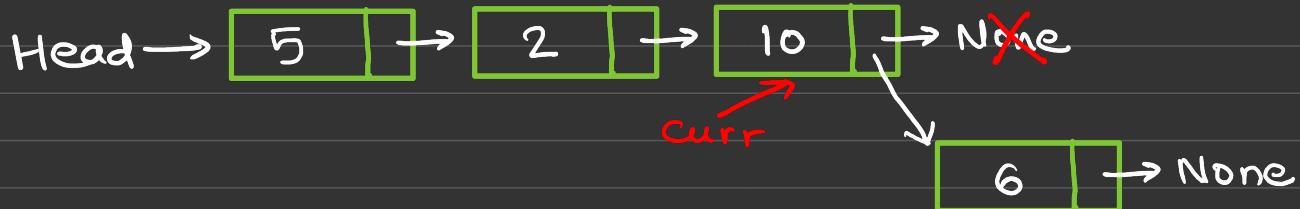
→ Iterate curr such that it points to the last node



→ Create an empty node (temp)

→ Add value(6) and next(None)

→ Make curr.next point to the temp.



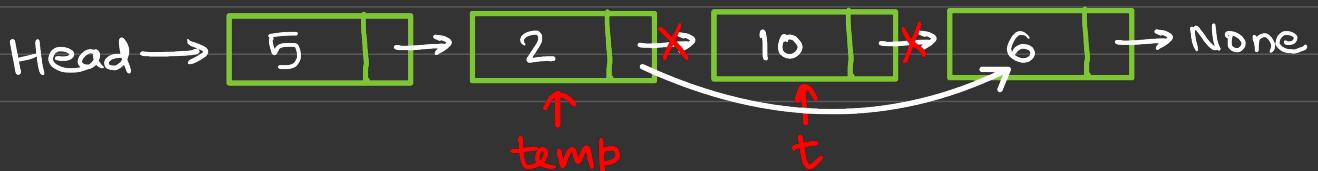
⑤ Delete(0) ← Delete in the beginning.

- Create a temp var(t) which references head.
- make head point to head.next
- free memory pointed by t

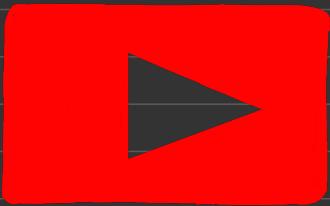


⑥ Delete(2) ← Delete in between/end

- make temp = head. and iterate temp to point $(i-1)$ th node
- make t = temp.next (node to be deleted)
- mark temp.next = t.next
- free the node t



Like



Subscribe