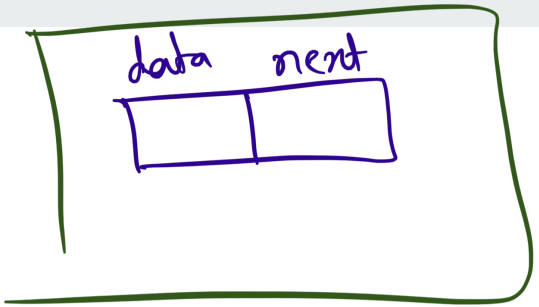
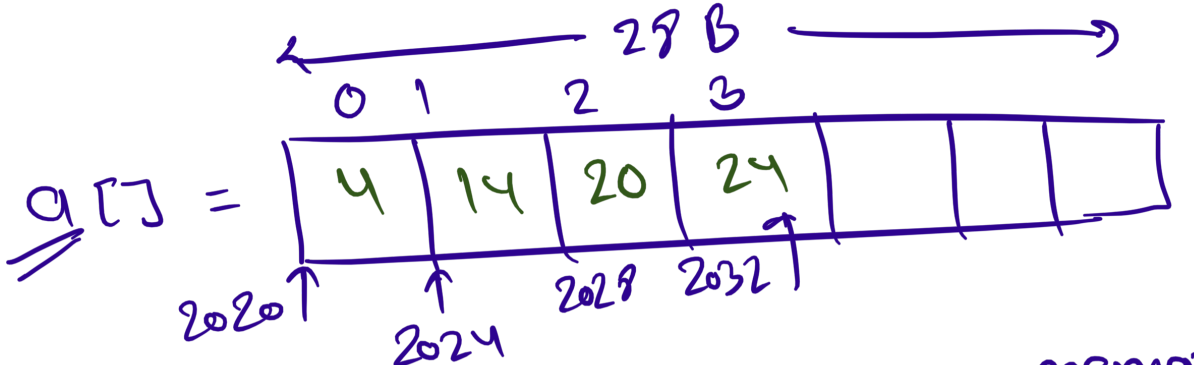
Linked List Basics



Linked List

A linked list is a linear data structure that includes a series of connected nodes. Here, each node stores the data and the address of the next node. For example,

Linear form data.



int → 4B

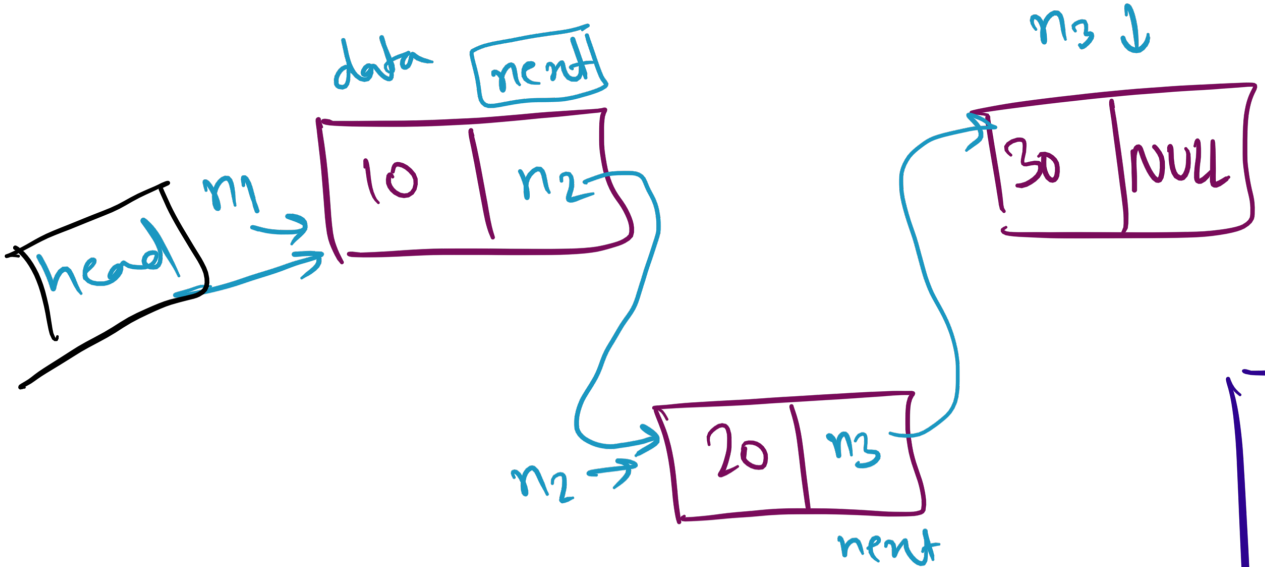
memory → 4×7 = 28 B

Contigous memory allocation.

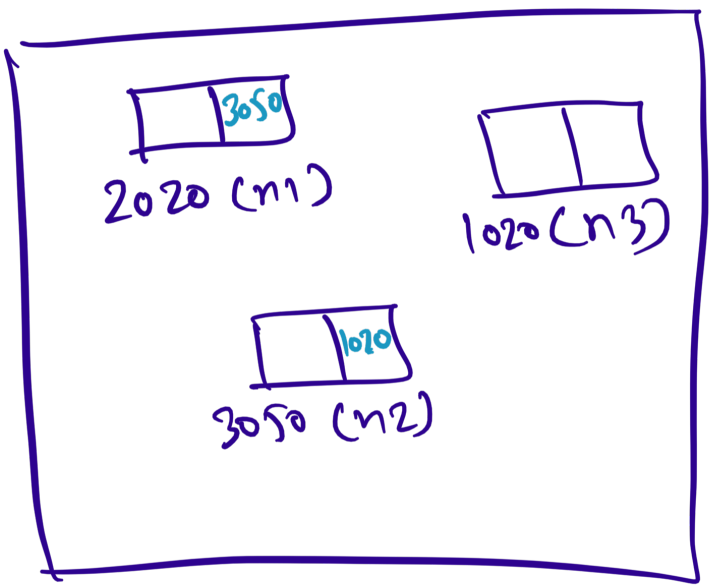
int a[] = new int[7];

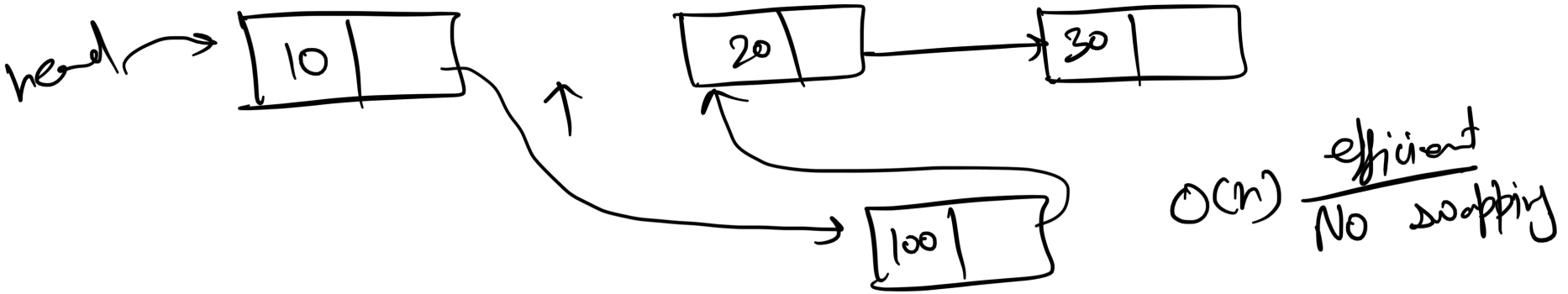
2020 + 3*4

2032

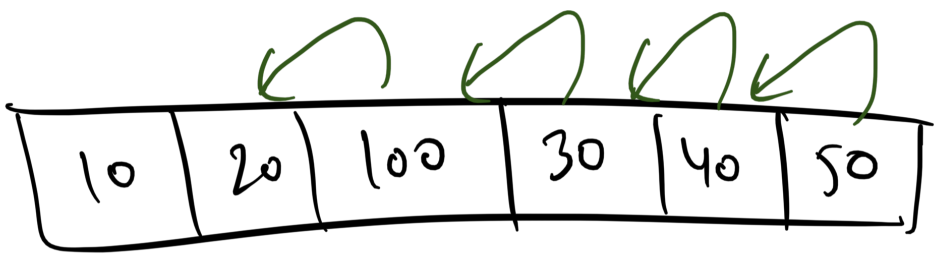



```
n1.next = n2;
n2.next = n3;
```



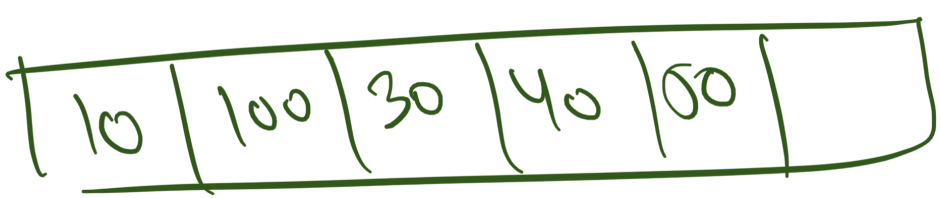


O(n) efficient
No swapping



insertion → O(n)

100

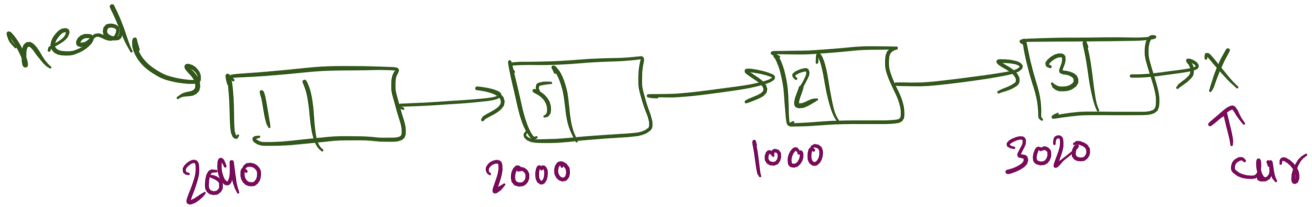


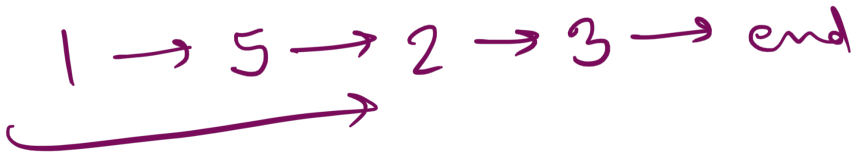
O(n)

Linked List vs Array

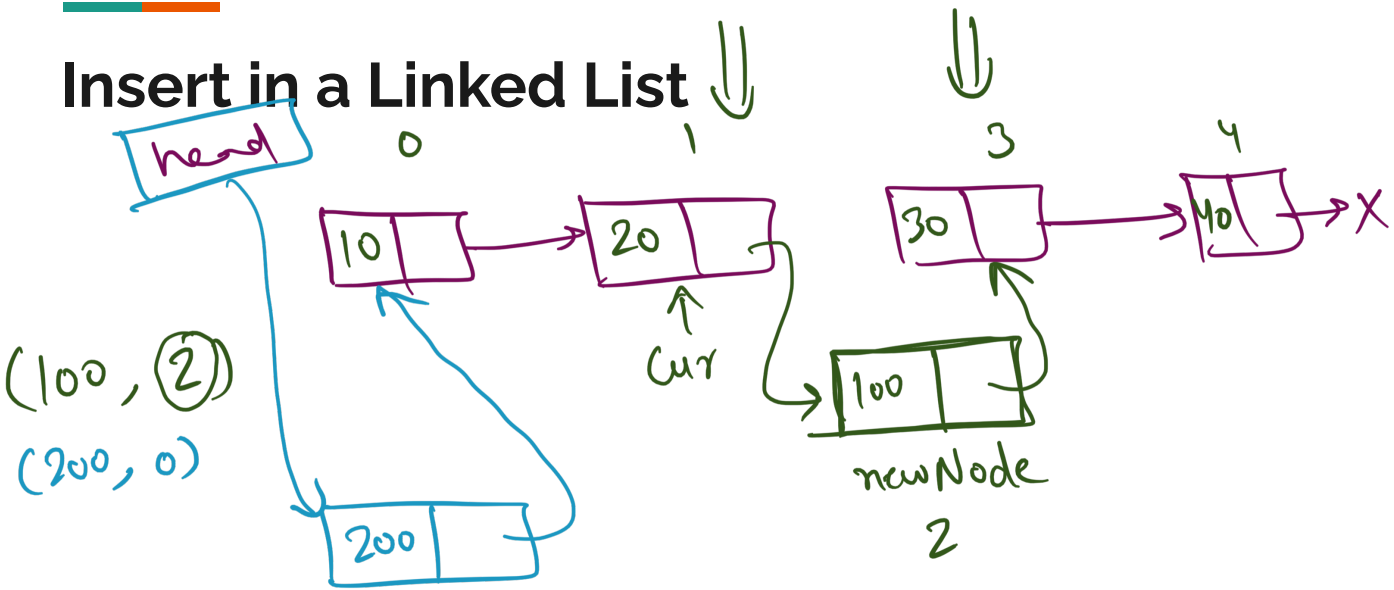
ARRAY	LINKED LISTS
1. Arrays are stored in contiguous location.	1. Linked lists are not stored in contiguous location.
2. Fixed in size.	2. Dynamic in size.
3. Memory is allocated at compile time.	3. Memory is allocated at run time.
4. Uses less memory than linked lists.	4. Uses more memory because it stores both data and the address of next node.
5. Elements can be accessed easily.	5. Element accessing requires the traversal of whole linked list.
6. Insertion and deletion operation takes time.	6. Insertion and deletion operation is faster.

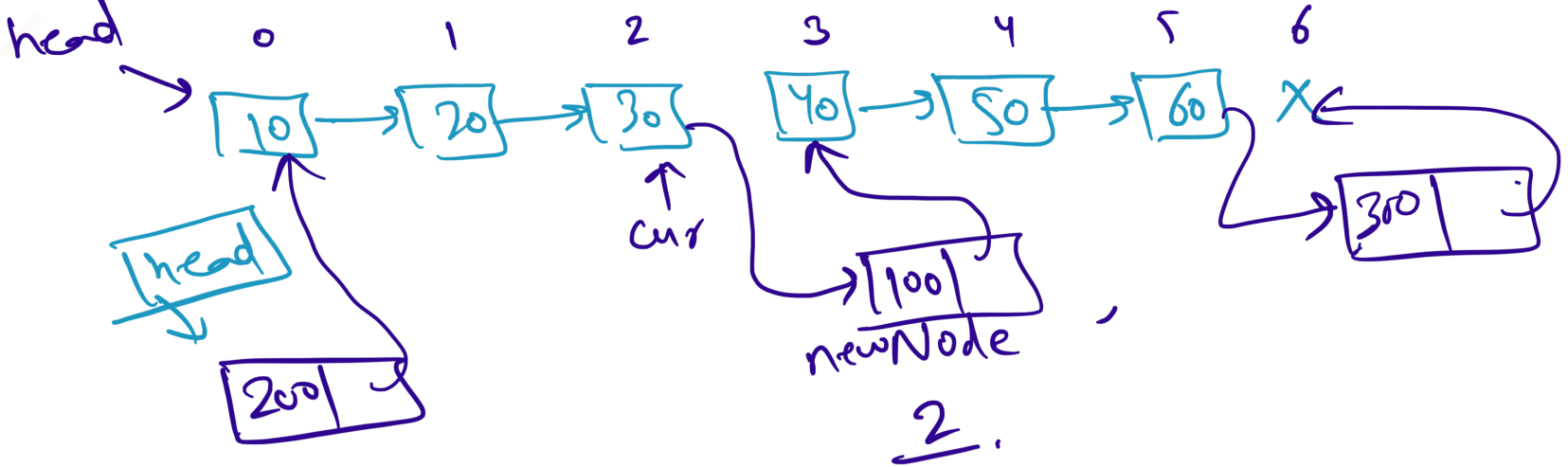
Traverse in a Linked List





Insert in a Linked List

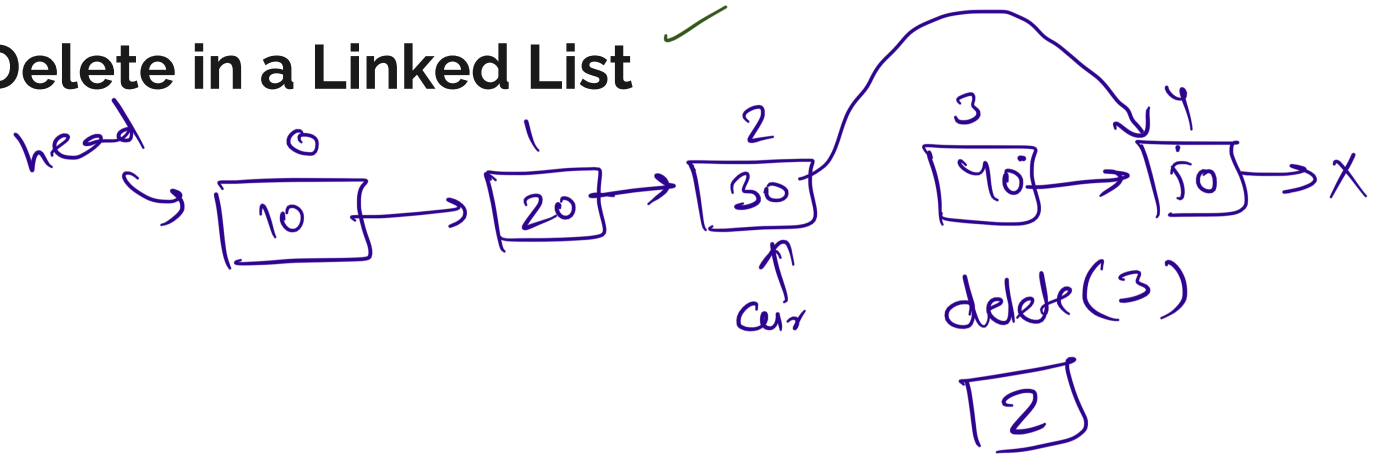




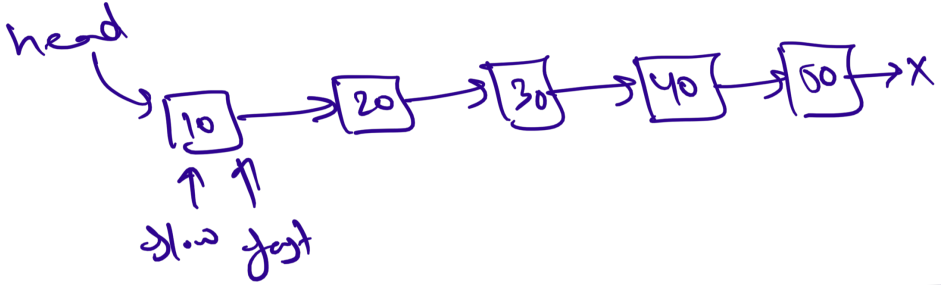
2.

```
main (head){
  head = insert( );
```

Delete in a Linked List



Find the Middle Element in a Linked List



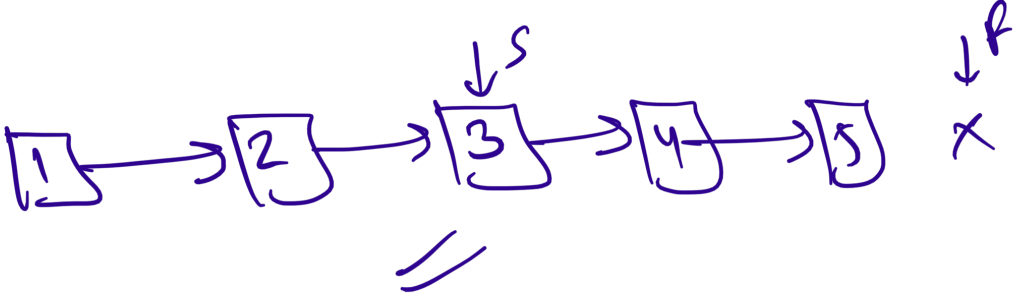
Count = 5

```
while (cur != null) {
    count++;
    cur = cur.next;
}
for (int i=0; i < count/2; i++)
```

2 - Traversals

1 - traversal

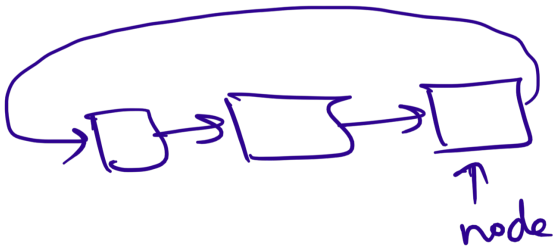
```
slow = slow.next;
fast = fast.next.next;
```

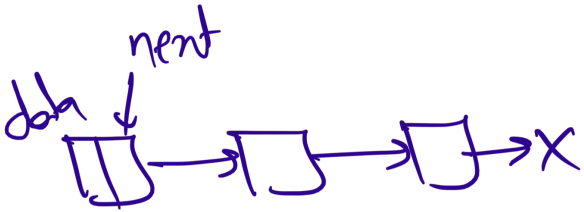


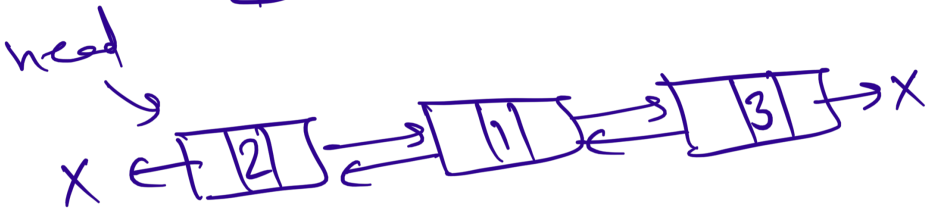
$$\frac{6}{2} = 3$$

Types of Linked Lists

1. Singly Linked List
2. Doubly Linked List
3. Circular Linked List

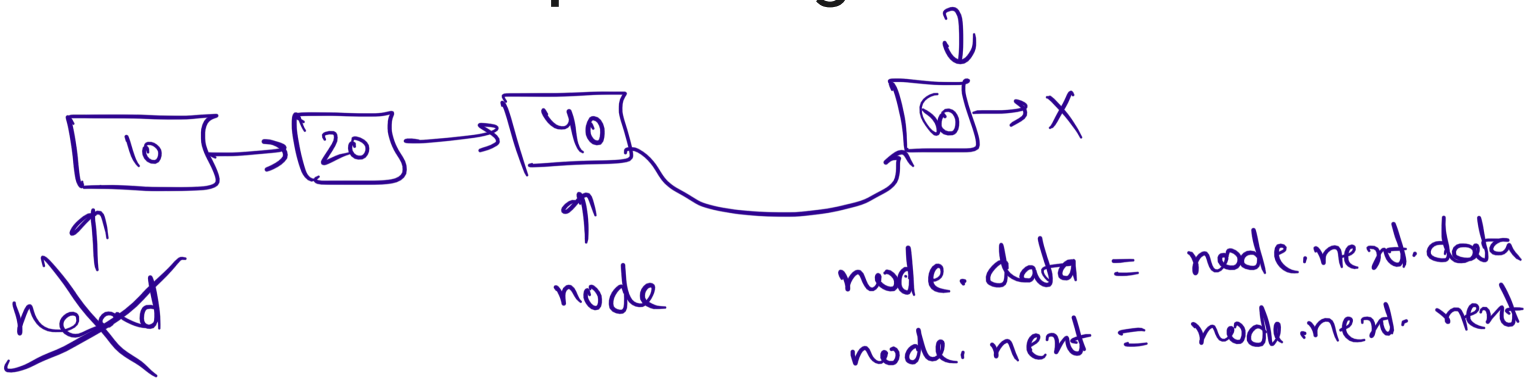







```
Node {
  Node next, Prev;
  int data;
}
```

Delete an element whose pointer is given in a Linked List.



Practice Problems

1. Find the Kth Element from last in a Linked List.
2. Remove duplicates from a Sorted Linked List.
3. Sort a Linked List using Bubble sort.
4. Find the intersection of Two sorted Linked List.
    - First linked list: 1->2->3->4->6
    - Second linked list be 2->4->6->8,
    - Output: 2->4->6.
5. Check if a Singly Linked List is a Palindrome.