

Doing the best at this moment, puts you in the best place for the next moment.



Introduction to Linked List



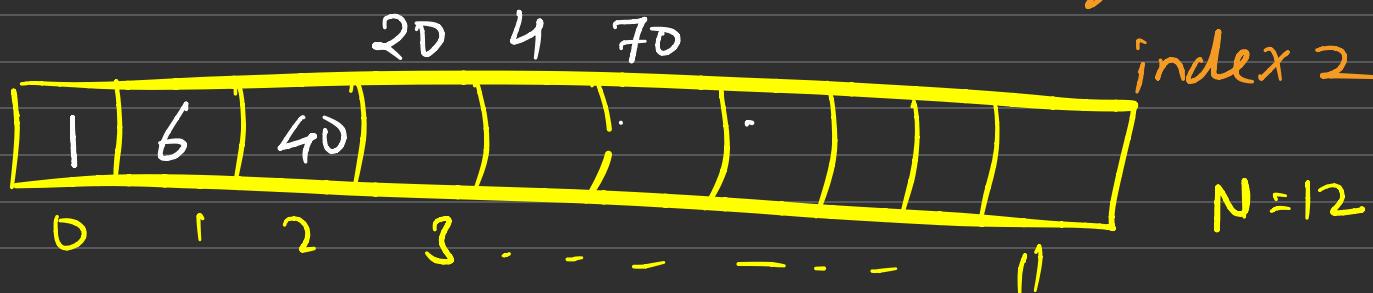
If we have Array, then Why Linked List

~~x = 20~~ int 3

insertion / deletion → difficult

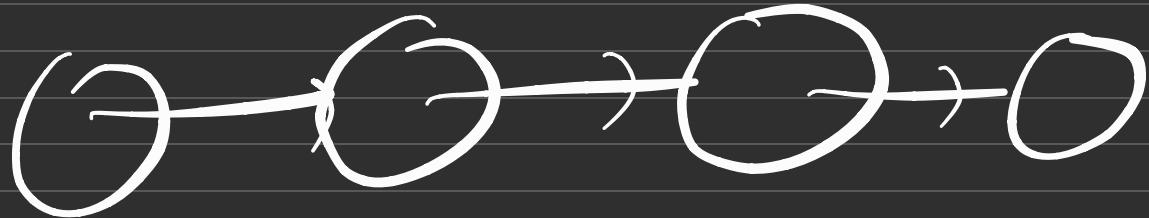
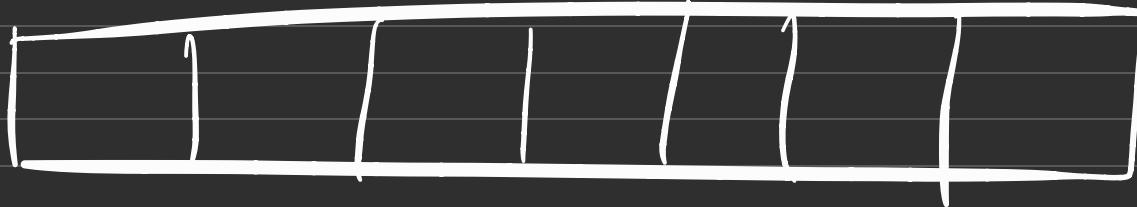
~~x = 70~~

x = 40 at
index 2





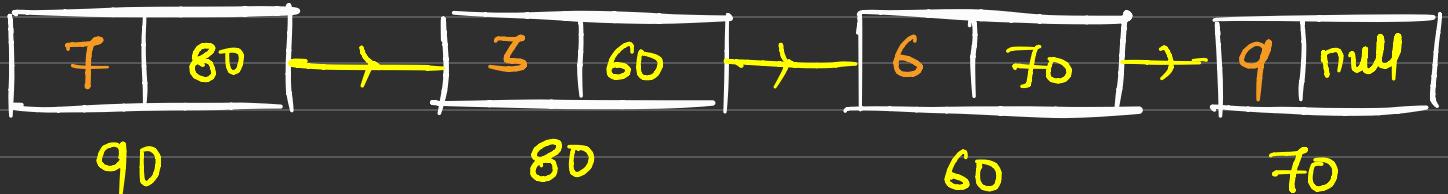
vector



What is Linked List



Collection of data which is represented in terms of Node



Pointer

Address of
next Node



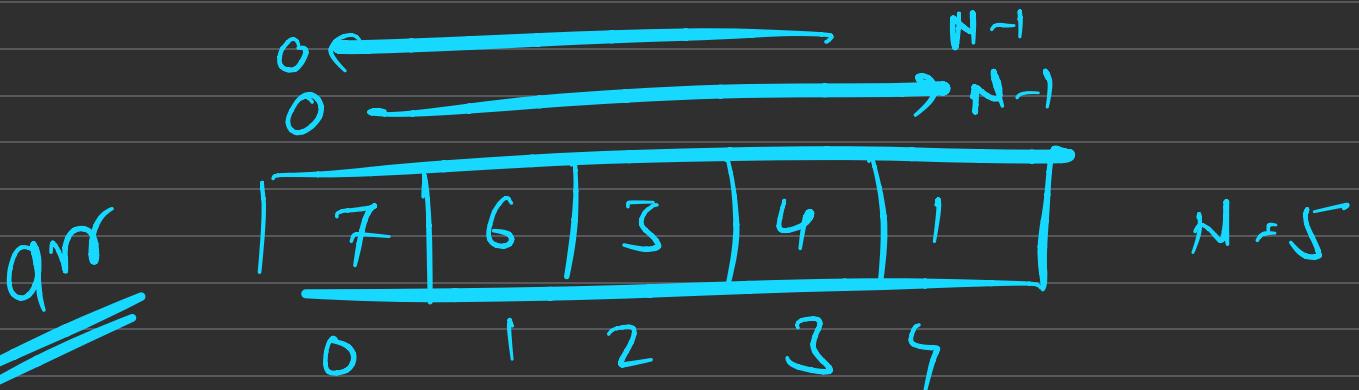
Data {

- string
- int
- char
- float
- double

Address {

- Address of next Node
- Null → if no next node

Head | Start Node





$\text{head} \rightarrow \text{start Node}$

$\text{if } \text{head} == \text{null} \rightarrow \text{list Empty}$

Not to change position of head

$\text{head} \Rightarrow \text{start of LL}$

Types of Linked List

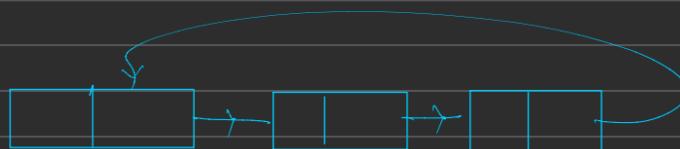
Singly Linked List



Doubly Linked List



Circular Linked List



- Types of operation:

* Insertion

- Start
- end
- Position

* Traversal

* Deletion

- start
- end
- Position

Structure of SLL

C++

```
class Node  
{  
    int data;  
    Node *next;  
};
```

Java

```
class Node  
{  
    int data;  
    Node next;  
};
```

Node



next
Node



fu

How to create Node



* new keyword → to allocate memory during RT

CPP = Node *node = new Node()

Java = Node node = new Node()

CPP

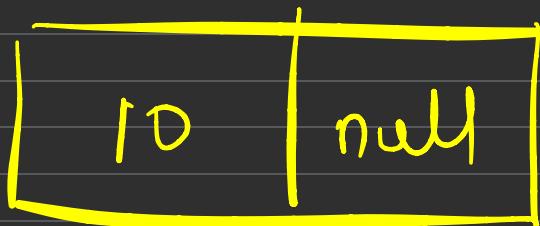
```
class Node  
{  
    int data  
    Node *next  
  
    Node (int data)  
    {  
        this->data = data  
        next = null  
    }  
}
```

Java

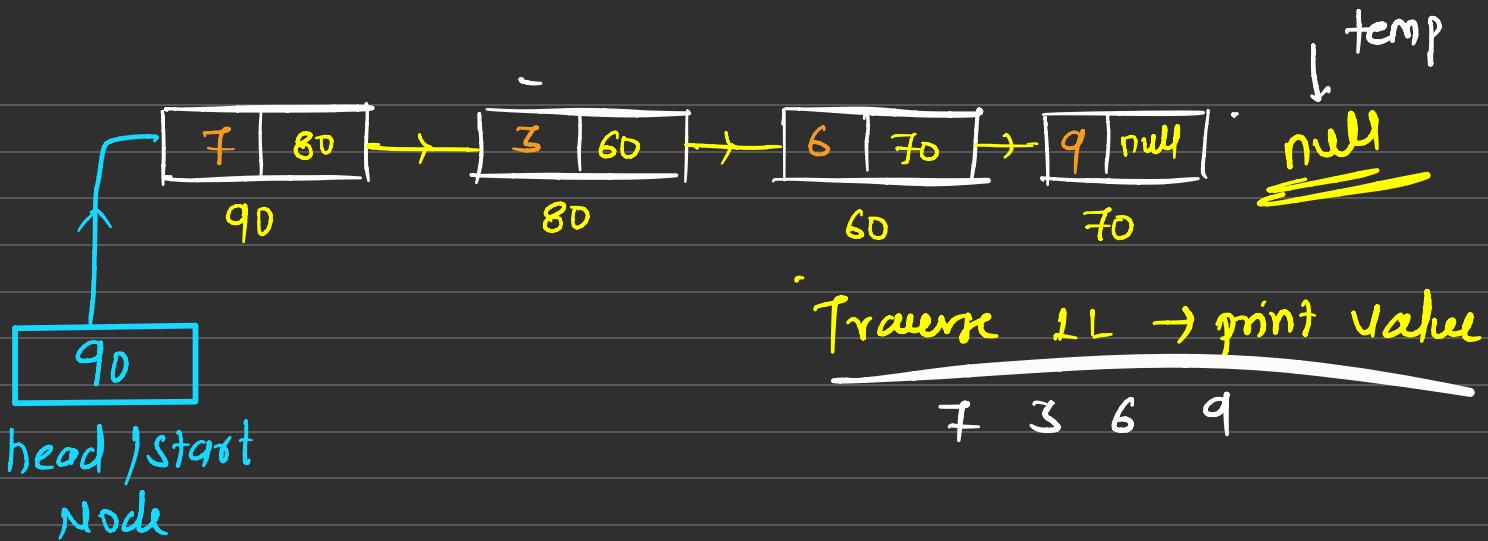
```
class Node  
{  
    int data  
    Node next  
  
    Node (int data)  
    {  
        this.data = data  
        next = null  
    }  
}
```

CPP = Node *node = new Node(10)

Java = Node node = new Node(10)



node
(700)



```

Node * temp = head
while ( temp != null )
    print ( temp -> data )
    temp = temp -> next
}

```

CPP

```

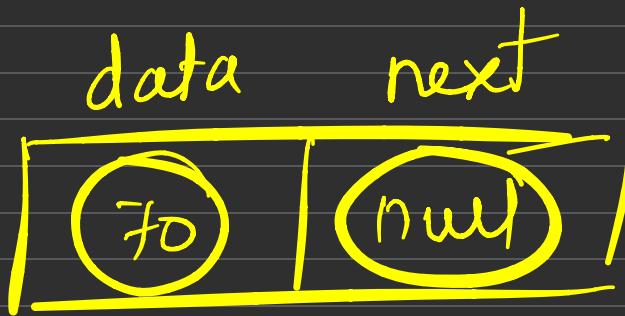
Node temp = head
while ( temp != null )
    print ( temp.data )
    temp = temp.next
}

```

Java

node->next
node->data

(&node).next
(*node).data



node
node->next
node->data

