DATA STRUCTURES AND ITS APPLICATIONS

Vandana M L

Department of Computer Science and Engineering

DATA STRUCTURES AND ITS APPLICATIONS

Doubly Linked List

Vandana M L

Department of Computer Science and Engineering

Doubly Linked List

- Linked list in which the nodes are linked together with two links which help to access predecessor and successor node from any node
- First node has no predecessor so contains a NULL pointer
- Last node has no successor so it has a NULL pointer

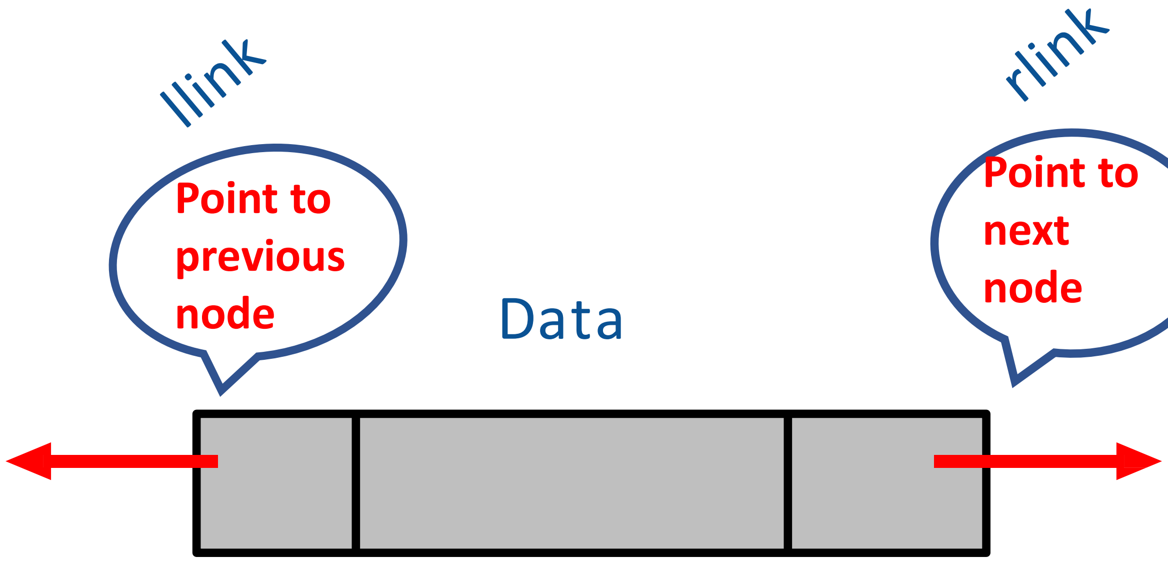
Doubly Linked List : Node Structure

A doubly linked list node contains three fields:

- Data
- link to the next node
- link to the previous node.

Doubly Linked List Node Structure

```
struct node
{
  int data;
  node*llink;
  node*rlink;
};
```



Doubly Linked List Vs Singly Linked List

Advantages:

- Can be traversed in either direction
- Some operations, such as deletion and inserting before a node, become easier

llink and rlink

Disadvantages:

- Requires more space
- List manipulations are slower Greater chance of having bugs

Two pointers

more links must be manipulated

DATA STRUCTURES AND ITS APPLICATIONS
Doubly Linked List Applications

Where we can use it??

Web Browser

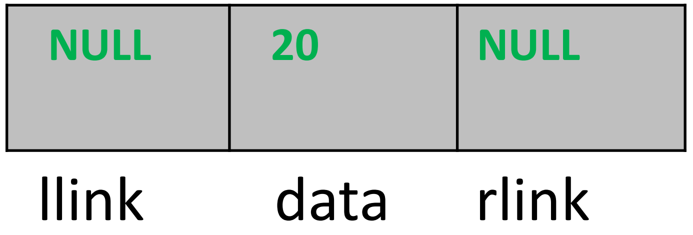
Music Player

Image Viewer

Creating a node

- Allocate memory for the node dynamically

- If the memory is allocated successfully
  set the data part to user defined value
  set the llink (address of previous node) and rlink (address of next node) part to NULL



DATA STRUCTURES AND ITS APPLICATIONS

Doubly Linked List Operations

Inserting a node

There are 3 cases

- Insertion at the beginning
- Insertion at the end
- Insertion at a given position

Insertion at the beginning

What all will change

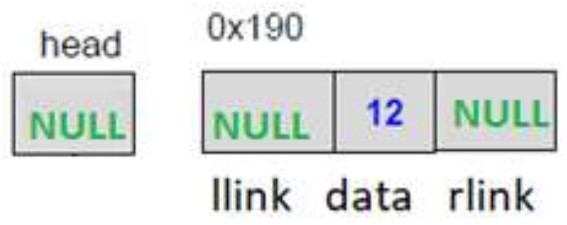
If the linked list empty(case 1)

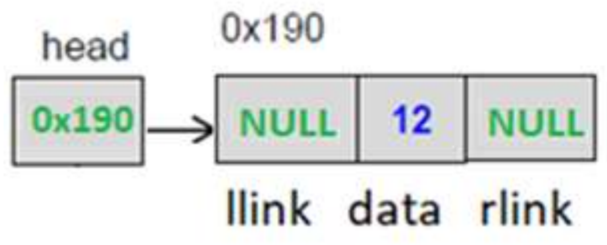
Head/Start pointer

else (case 2)

- Head/Start pointer
- New front's llink and rlink
- Old front's llink

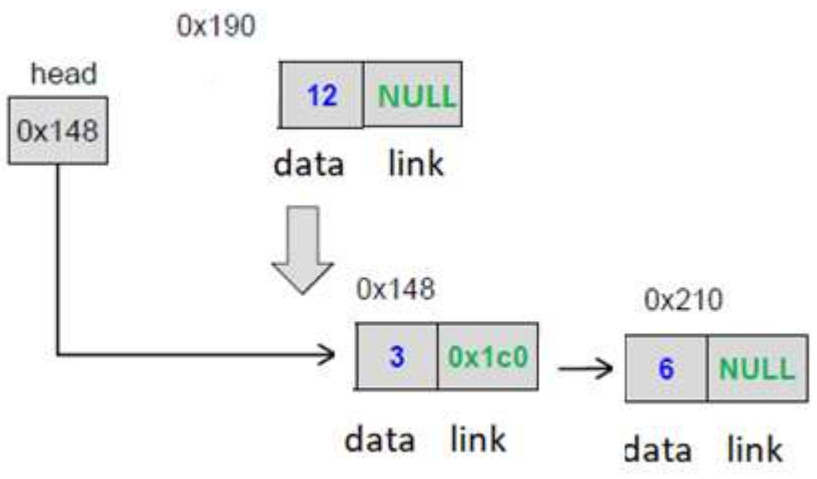
Insertion at the beginning (Case1)

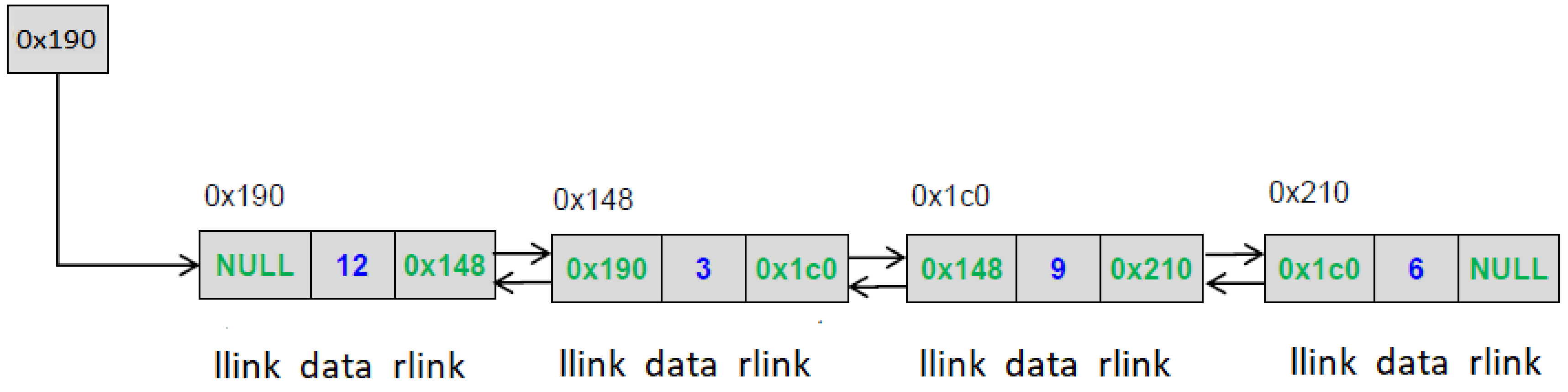




Doubly Linked List Operations

Insertion at the beginning(Case 2)





Insertion at the end

What all will change

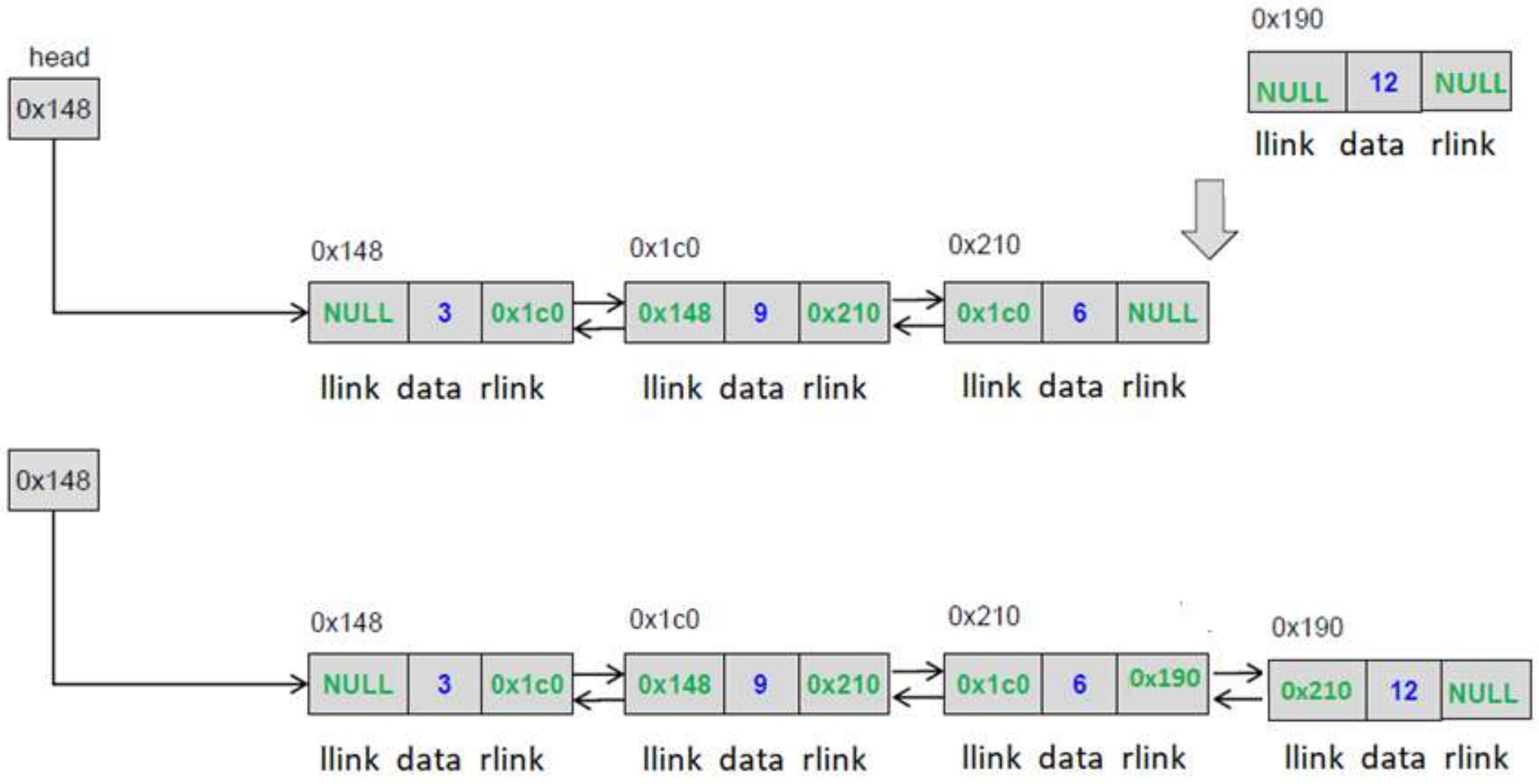
If the linked list empty(same as case 1 of insert at front)

    Head/Start pointer(case 2)

else

- ➢ Last node’s rlink
- ➢ New node’s llink

Insertion at the end

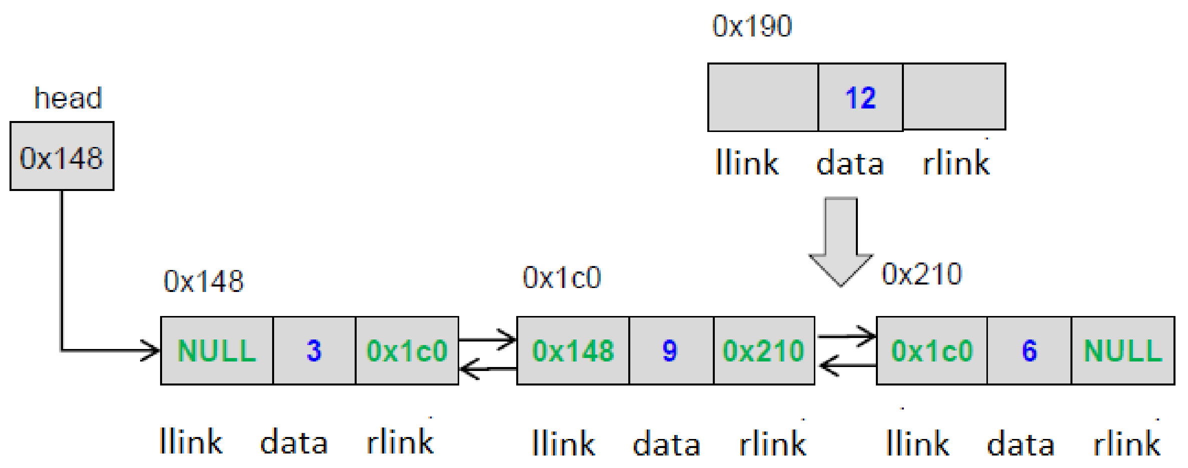


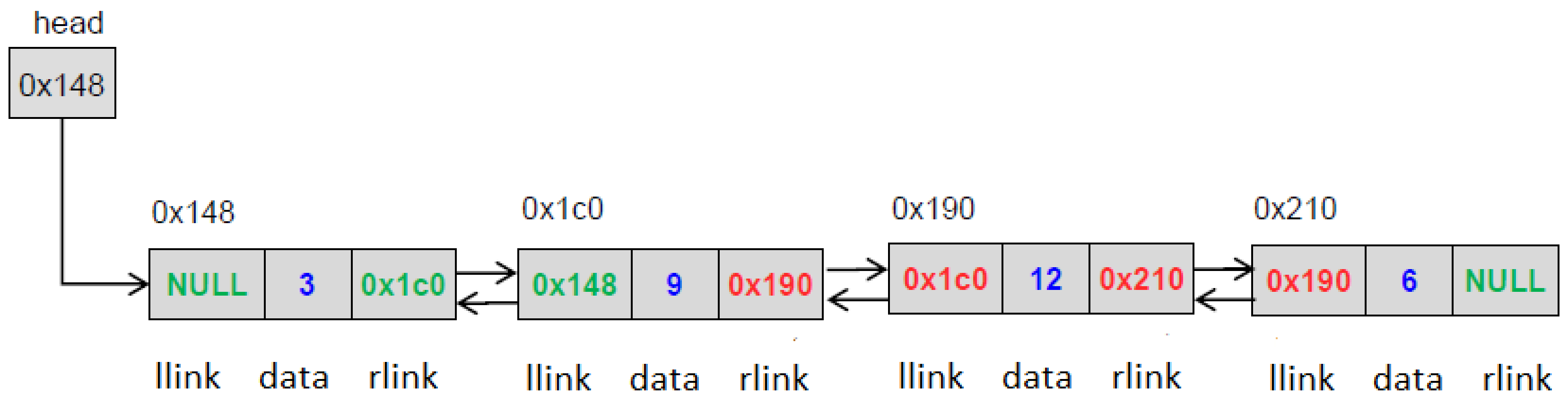
Insertion at the given position

```
➢ Create a node
If the list is empty or position is 1
    ➢ make the start pointer  point towards the new node;
Else
    ➢ Traverse the linked list to reach given position
    ➢ Keep track of the previous node
    If it is an intermediate position
        ➢ Change previous node rlink to point to the newnode
        ➢ Newnode's llink  to point to previous node and rlink  to point to  the next node
        ➢ Next node llink to point to the newnode
    Else
        if last position
        ➢       insert at the end
        Else
        invalid position
```

Doubly Linked List Implementation

Insertion at the given position





Doubly Linked List insert operation

Apply the concepts to implement following operations for a Doubly linked list

- Find the node pairs with a given sum in a doubly linked list
- Insert a node after a node with a given value
-

THANK YOU

Vandana M L

Department of Computer Science & Engineering

vandanamd@pes.edu

+91 7411716615