

Data Structures and Algorithms

Lecture 9: Linked List – Insertion

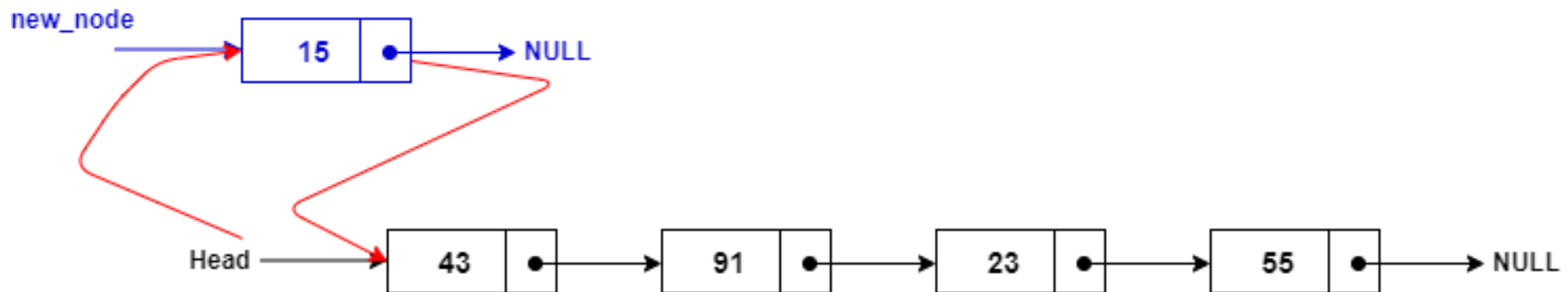
Linked List - Insertion

Inserting a node is one of the most fundamental linked list operations.

- At the front of the list
- At the end of the list
- Before a specified node
- After a specified node

Insert a Node at the Front of a Linked List

- Linked list head always points to the first node if there is any.
- It points to NULL for an empty list.
- So inserting a new node means the head will point to the newly inserted node.
- And the new node will point to where head was pointing before insertion.



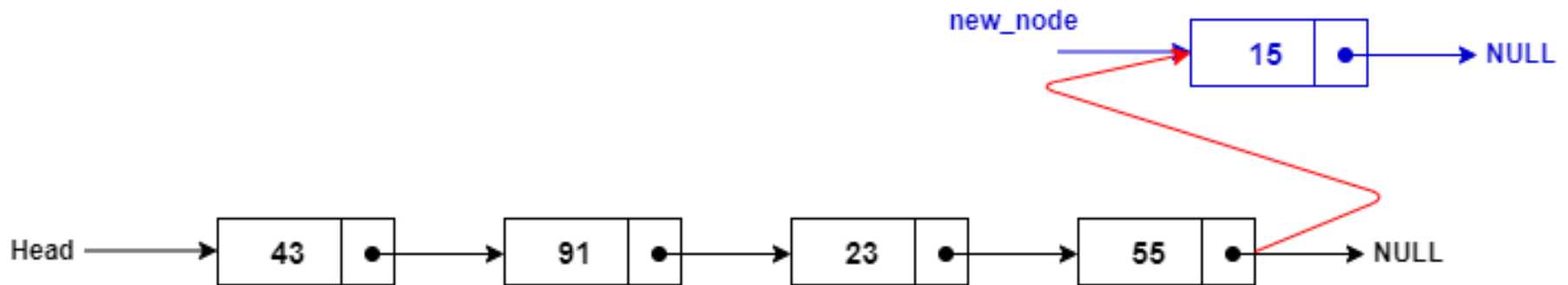
Insert a Node at the Front of a Linked List

- Insert at 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

```
struct node *newNode;  
  
newNode = malloc(sizeof(struct node));  
  
newNode->data = 15;  
  
newNode->next = head;  
  
head = newNode;
```

Insert a Node at the End of a Linked List

- The last node of linked list always points to NULL.
- To insert a new node at the end of the list, you have to make the current last node point to the new node.
- And the new node will point to NULL. So the newly inserted node will become the last node after insertion.



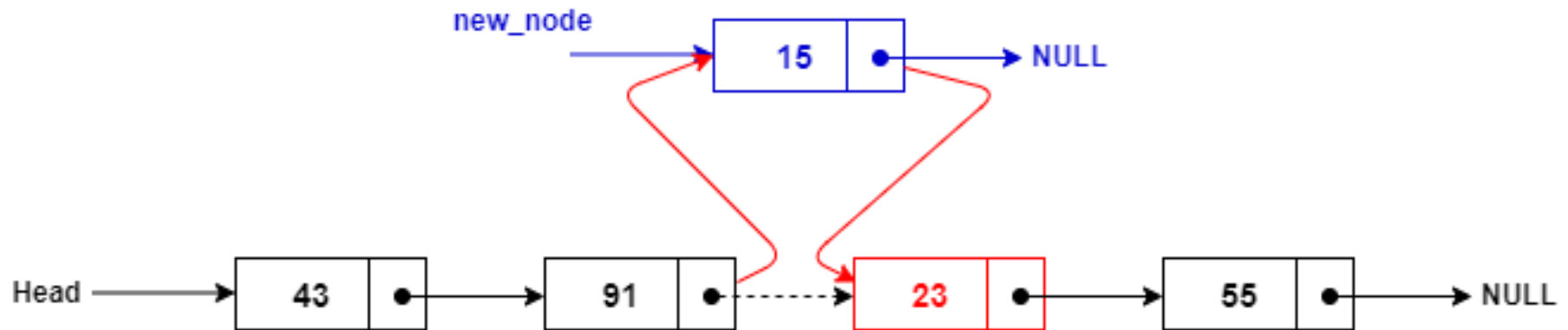
Insert a Node at the End of a Linked List

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

```
struct node *newNode;  
  
newNode = malloc(sizeof(struct node));  
  
newNode->data = 15;  
  
newNode->next = NULL;  
  
struct node *temp = head;  
  
while(temp->next != NULL)  
  
    temp = temp->next;  
  
temp->next = newNode;
```

Insert the New Node at a Specified Node

- The newly inserted node will become the next node a specified node.
- The node will be specified by the value of the node here.



Insert the New Node at a Specified Node

- 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

```
struct node *newNode;  
newNode = malloc(sizeof(struct  
node));  
newNode->data = 15;  
struct node *temp = head;  
for(int i=2; i < position; i++)  
{  
    if(temp->next != NULL)  
        temp = temp->next;  
}  
newNode->next = temp->next;  
temp->next = newNode;
```


Recall:

- Linked list – Insertion at the beginning, end, at a specified point.

