Data Structures and Algorithms

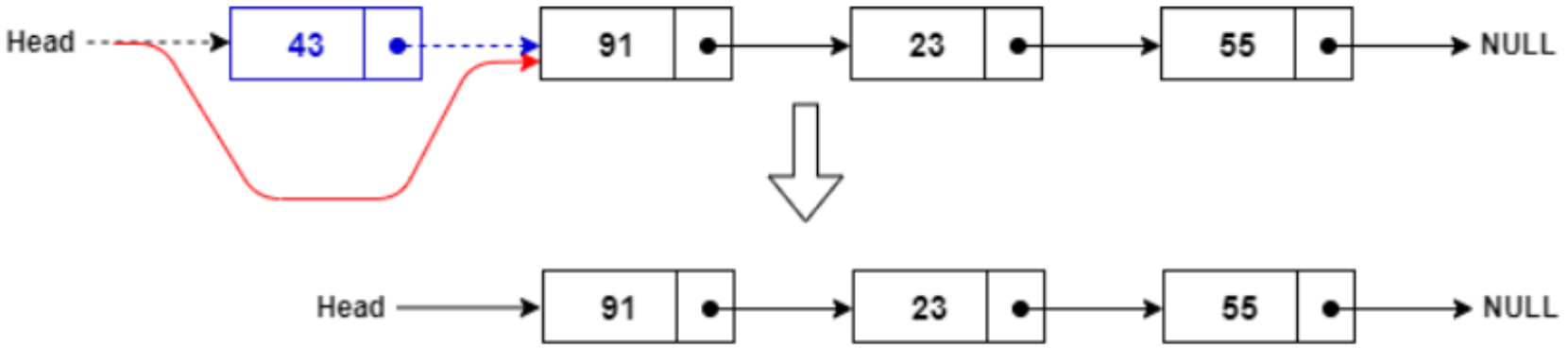
Lecture 10: Linked List – Deletion

Linked List - Deletion

- Head always points to the first node.
- Head will move to where the first node is currently pointing to.
- First node can either point to the second node or to NULL in case the first node is the only node in the linked list.
- After moving the head, we can free up the memory of the first node.

Delete First Node of a Linked List

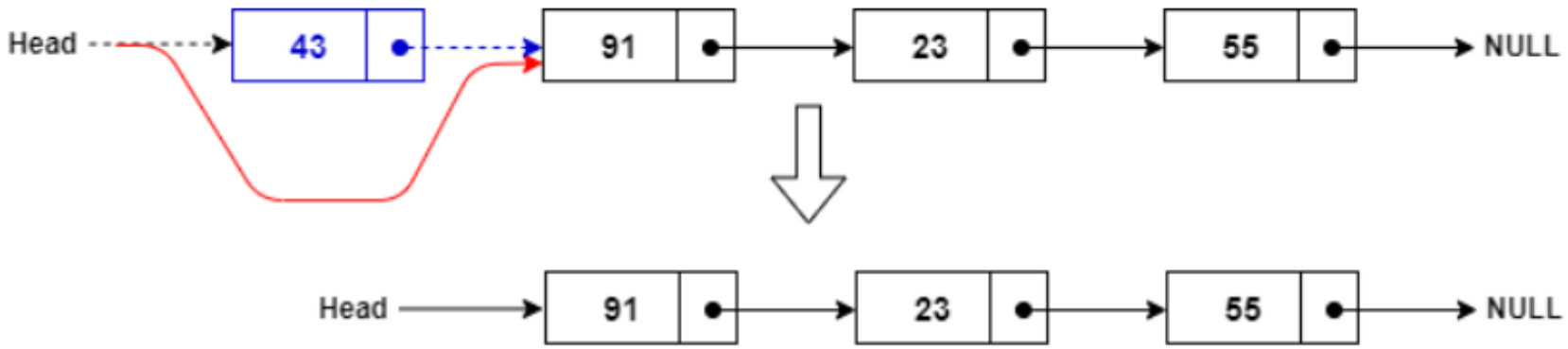
- If head is NULL, return. There is no element in the linked list.
- Assign the head pointer to a temporary variable, tmp.
- Move the head to the next node. If the linked list has only one node, head will move to NULL.
- Delete the temporary pointer, tmp. As tmp is pointing to the first node, first node will be deleted.



Delete First Node of a Linked List

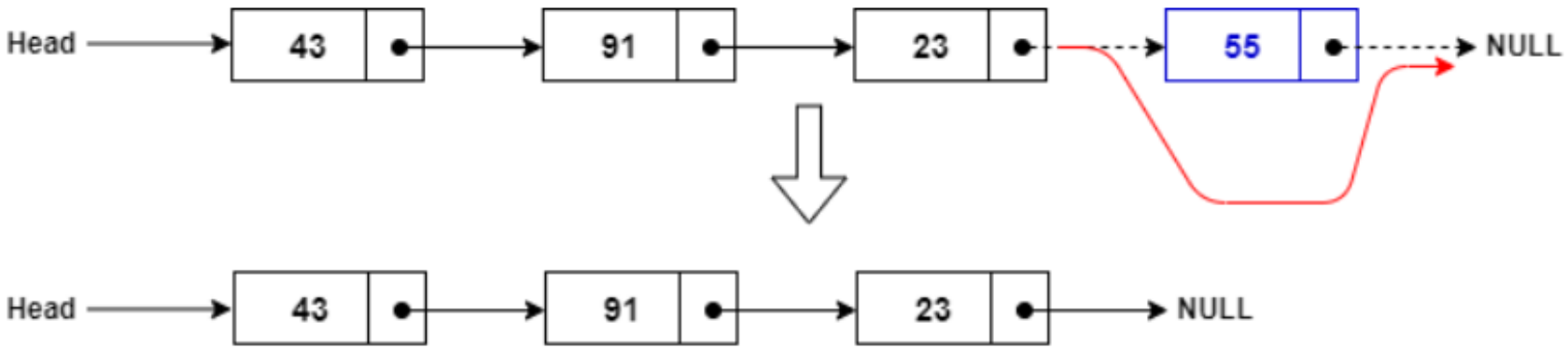
- Point head to the second node

```
head = head->next;
```



Delete from End in Linked List

- The last node always points to NULL.
- To delete that, we have to traverse up to last node.
- We have to point the second last node to NULL and free memory of the last node.
- After deletion the second last node will become the new last node.



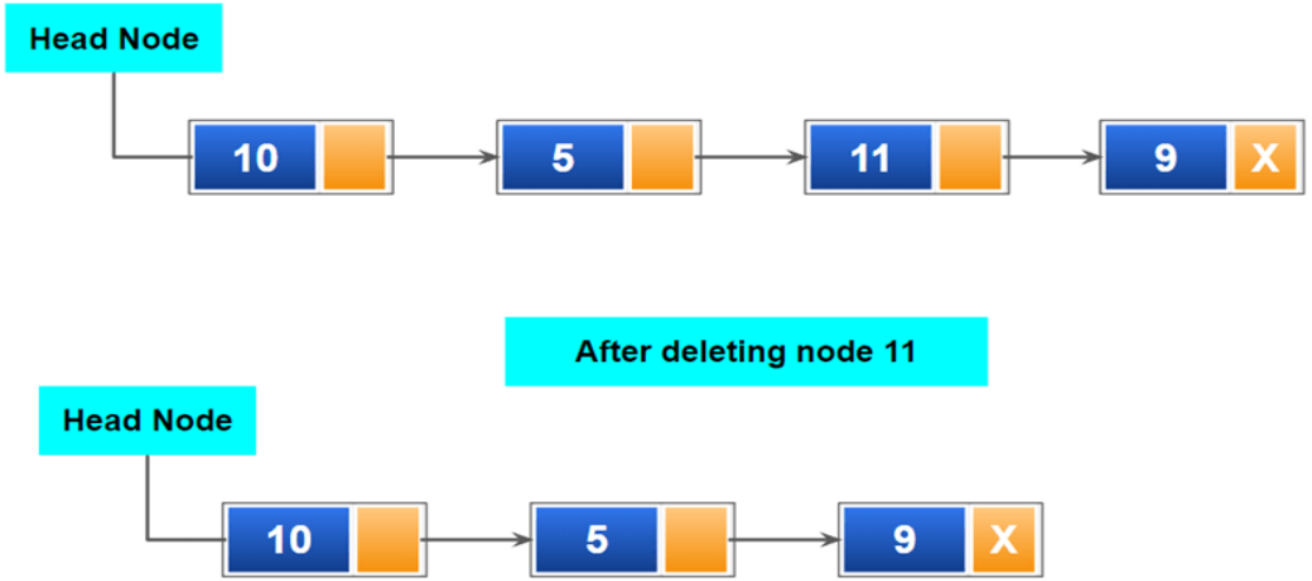
Delete from End in Linked List

- Traverse to second last element
- Change its next pointer to null

```
struct node* temp = head;
while(temp->next->next!=NULL)
{
  temp = temp->next;
}
temp->next = NULL;
```

Delete from Middle in Linked List

- Traverse until the end of the linked list. Check if the current node is equal to the search node or not.
- If any node matches, store the node pointer to the current node.
- The “next” of the previous node will be the next node of the current node.
- Delete and free the memory of the current node.



Delete from Middle in Linked List

- Traverse to element before the element to be deleted
- Change next pointers to exclude the node from the chain

```
for(int i=2; i< position; i++)
{
  if(temp->next!=NULL)
  {
    temp = temp->next;
   }
}
temp->next = temp->next->next;
```

Recall:

- Linked list – Deletion at the beginning, end, at a specified node.