CSCI 2270 – Data Structures and Algorithms

Instructor: Hoenigman/Zagrodzki/Zietz

Recitation 4

Objectives:

1. What is a Linked List
2. Insertion
3. Deletion
4. Printing
5. Recitation Exercise

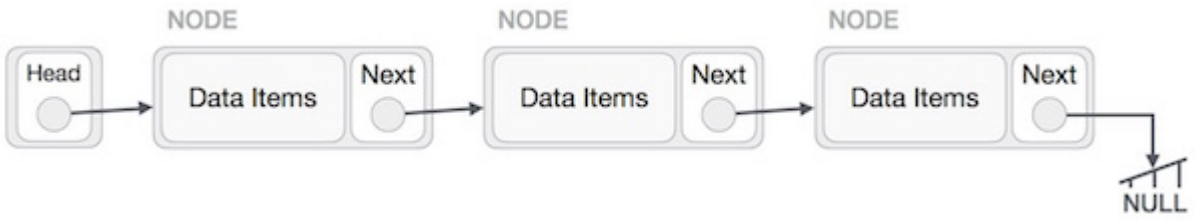
What is a Linked List?

A linked list is a sequence of data structures, which are connected via links.

Let's elaborate:

Linked List is a sequence of links which contains items. Each link contains a connection to another link. Linked list is the second most-used data structure after array. Following are the important terms to understand the concept of Linked List.

- **Node** − Each Node of a linked list can store data and a link.
- **Link** − Each Node of a linked list contains a link to the next Node (unit of Linked List) called Next.
- **LinkedList** − A Linked List contains a connection link from the first link called Head. Every Link after the head points to another Node (unit of Linked List). The last node points to NULL.

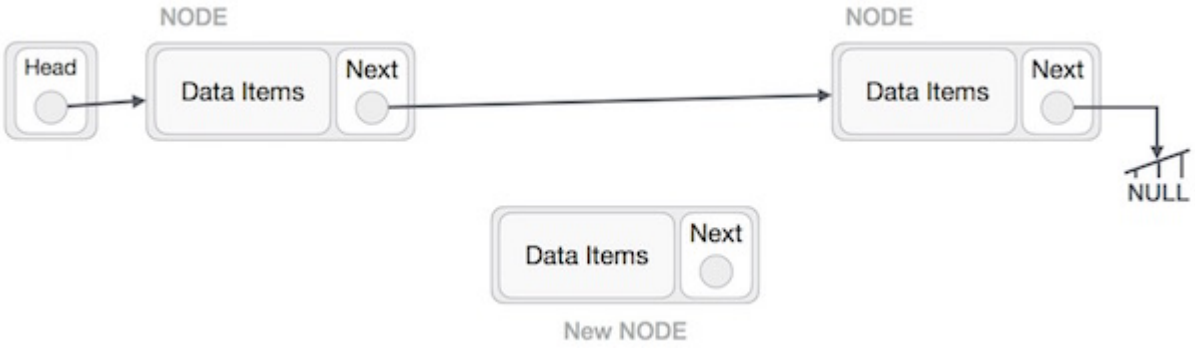


Insertion

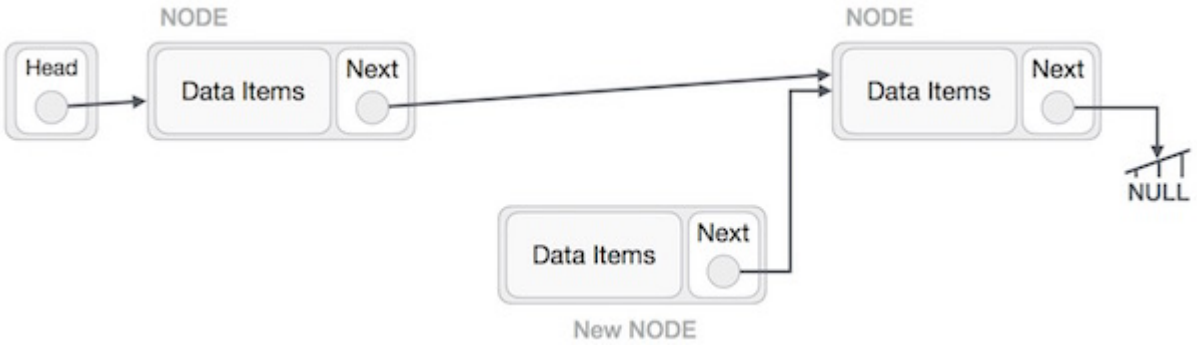
Adding a new node in a linked list is a multi-step activity. We shall learn this with diagrams here. First, create a node using the same structure and find the location where it must be inserted.

Let us take an example, where we are inserting a node B (NewNode), between A (LeftNode) and C (RightNode).

Step 1:

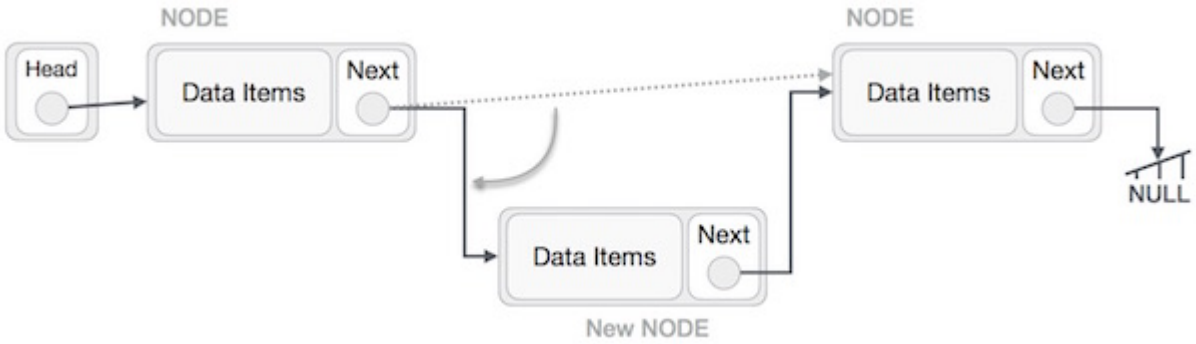


Step 2:



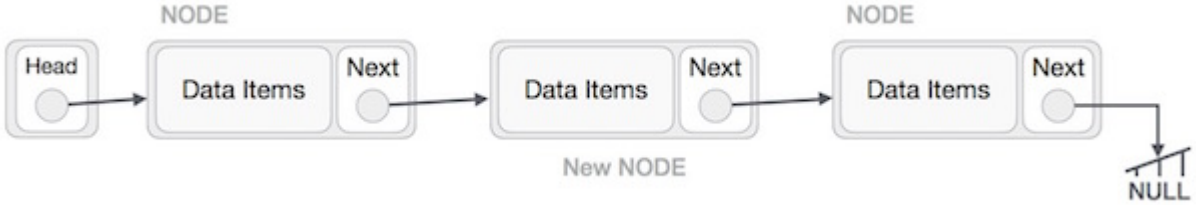
NewNode.next −> RightNode;

Step 3:



LeftNode.next −> NewNode;

Step 4:



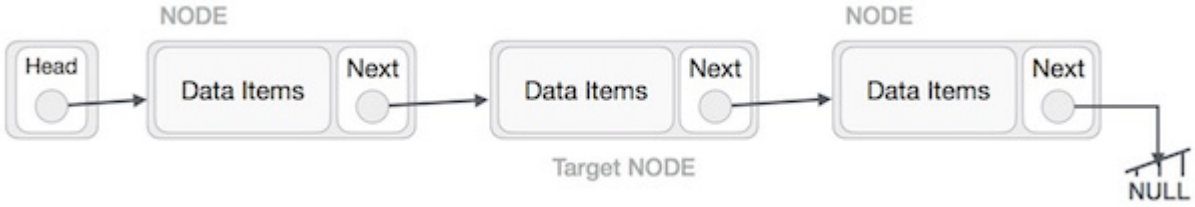
Now, we have put the new node in the middle of the two nodes. The new list should look like the figure shown above in Step 4.

Similar steps should be taken if the node is being inserted at the beginning of the list. While inserting it at the end, the second last node of the list should point to the new node and the new node will point to NULL.

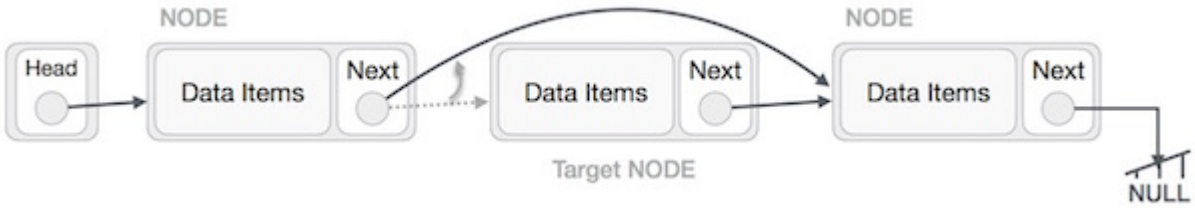
Deletion

Deletion is also a multi-step process. First, locate the target node to be removed, by using searching algorithms (and finding the required key).

Step1:



Step2:



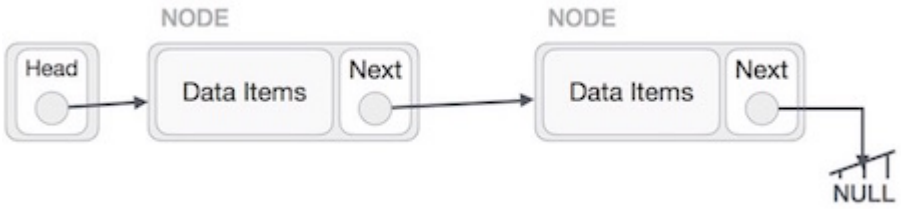
LeftNode.next −> TargetNode.next;

This will remove the link that was pointing to the target node.

Step 3:

TargetNode.next −> NULL;

Step 4:



If we need to use the deleted node, we can keep it in memory. Otherwise we can simply deallocate memory and wipe off the deleted node completely.

Printing

To print a list, we need to traverse through all the nodes in the list until we encounter the last node. When a node points to NULL we know that it is the last node.

Code Snippet:

```
node = root;
while ( node != NULL ) {
  cout<< node->value;
  node = node->next;
}
```

Recitation Exercise

Write a program to insert a node in the linked list. The only criterion for insertion is encountering an “even key”. Thus, only insert a node when you encounter an even key in the linked list.

Finally, print 2 lists, first the list without the insertions and then the list with the insertions.