Singly Linked List Implementation

BY [SJ](http://algorithms.tutorialhorizon.com/author/sumitjain/) · AUGUST 12, 2014

Linked List– As the name sug­gests it’s a list which is linked.

* + Linked List con­sist of Nodes

Nodes are noth­ing but objects of a class and each node has data and a link to the next node.

class Node {

public int data;

public Node next;

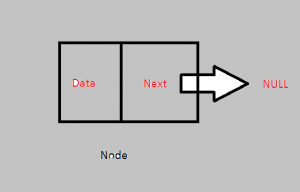
public Node(int data) {

this.data = data;

this.next = null;

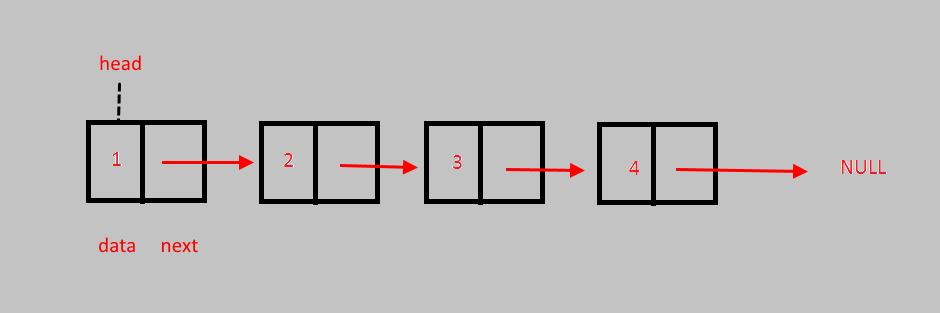
}

}

[](http://algorithms.tutorialhorizon.com/files/2014/08/Node.png)

*Node*

* The last node in the list points to NULL , so when you reach there you will know that the list ends here.

[](http://algorithms.tutorialhorizon.com/files/2014/08/Linked-List.png)

*Linked List*

**Oper­a­tions:**

**Add at the Start :** Add a node the begin­ning of the linked list. Its O(1).

**Add at the End :** Add a node at the end of the linked list. its O(n) since to add a node at the end you need to go till the end of the array.

**Delete at the Start :** Delete a node from begin­ning of the linked list. Its O(1).

**Delete at the End :** Delete a node from the end of the linked list. its O(n) since to delete a node at the end you need to go till the end of the array.

**Get Size:** returns the size of the linked list.

**Get Ele­ment at Index :** Return the ele­ment at spe­cific index, if index is greater than the size then return –1. its O(n) in worst case.

**Add Ele­ment at Spe­cific Index :**Add ele­ment at spe­cific index. If index is greater than size then print “INVALID POSITION”. Worst case its O(n)

**Dis­play():** Prints the entire linked list. O(n).