

AP ASSIGNMENT

Name: Swatantra

UID: 22BCS12725

Section: 612-“B”

Print linked list – GFG

```
class Solution {
public:
    // Function to display the elements of a linked list in the same line without trailing space
    void printList(Node *head) {
        Node* temp = head;
        while (temp != nullptr) {
            cout<< temp->data;
            if (temp->next != nullptr) {
                cout<< " ";
            }
            temp = temp->next;
        }
        cout<<endl; }
};
```

The screenshot shows a C++ IDE with two panes. The left pane, titled 'Output Window', has a sub-tab 'Compilation Results' showing 'Compilation Completed'. Below this, it shows 'For Input: 12', 'Your Output: 12', and 'Expected Output: 12'. The right pane shows C++ code for a linked list. It includes a 'Node' struct, a 'Node' constructor, and a 'Solution' class with a 'printList' function. The code is as follows:

```
1 // Driver Code Ends
19 /*
20  int data;
21  struct Node {
22      struct Node* next;
23  }
24  Node(int x) {
25      data = x;
26      next = nullptr;
27  }
28 };
29 */
30 /*
31  Print elements of a linked list on console
32  Head pointer input could be NULL as well for empty list
33  */
34
35 class Solution {
36 public:
37     // Function to display the elements of a linked list in the same line without trail:
38     void printList(Node *head) {
39         Node* temp = head;
40         while (temp != nullptr) {
41             cout << temp->data;
42             if (temp->next != nullptr) {
43                 cout << " ";
44             }
45             temp = temp->next;
46         }
47         cout << endl;
48     }
49 };
50 // Driver Code Ends
```

Remove duplicates from a linkedlist

```
class Solution {
public ListNode deleteDuplicates(ListNode head) {
    ListNode current = head;

    while (current != null && current.next != null) {
        if (current.val == current.next.val) {
            current.next = current.next.next;
        } else {
            current = current.next;
        }
    }

    return head;
}
```

Accepted 168 / 168 testcases passed

Swatantra Yadav submitted at Feb 16, 2025 20:03

Editorial

Solution

Runtime

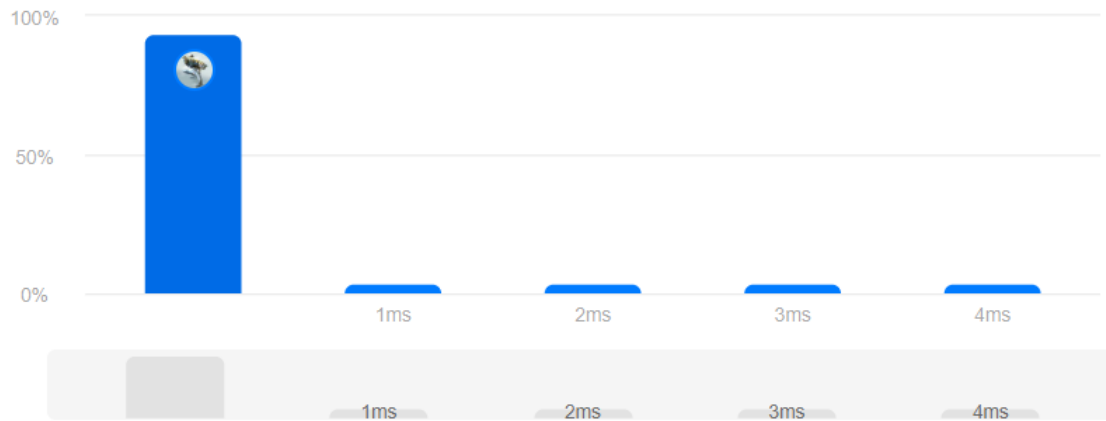


0 ms | Beats 100.00% 🌿

[Analyze Complexity](#)

Memory

44.16 MB | Beats 55.12% 🌿



Reverse a linked list


```
class Solution {  
    public ListNode reverseList(ListNode head) {  
        ListNode node = null;  
  
        while (head != null) {  
            ListNode temp = head.next;  
            head.next = node;  
            node = head;  
            head = temp;  
        }  
  
        return node;  
    }  
}
```

```

    }
} return prev;
}


```

Accepted 28 / 28 testcases passed

 **Swatantra Yadav** submitted at Feb 16, 2025 20:04

 Editorial

 Solution


 Runtime

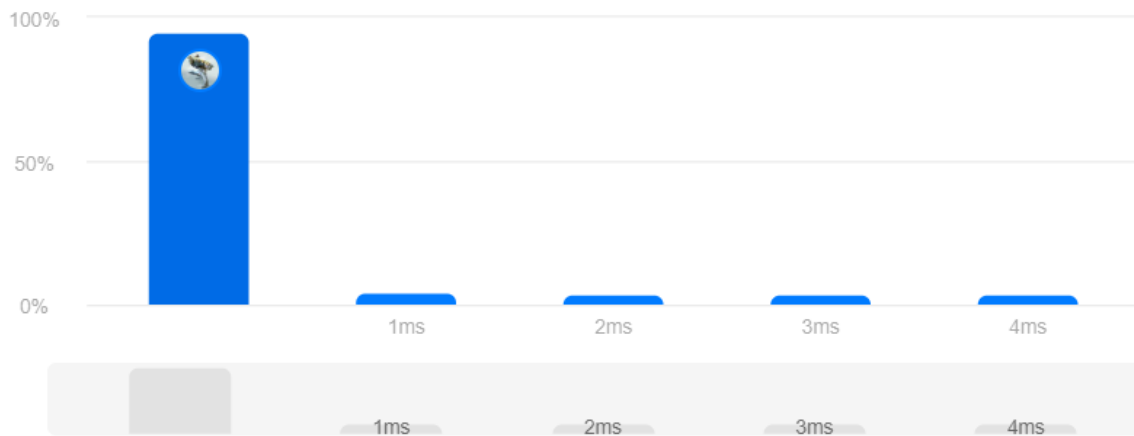


0 ms | Beats **100.00%** 

 [Analyze Complexity](#)

 Memory

42.28 MB | Beats **91.52%** 



Delete middle node of a list


```


class Solution {
    public ListNode middleNode(ListNode head) {
        ListNode slow = head;
        ListNode fast = head;
        while(fast!=null && fast.next!=null){
            slow = slow.next;
            fast = fast.next.next;
        }
        return slow;
    }
}

```

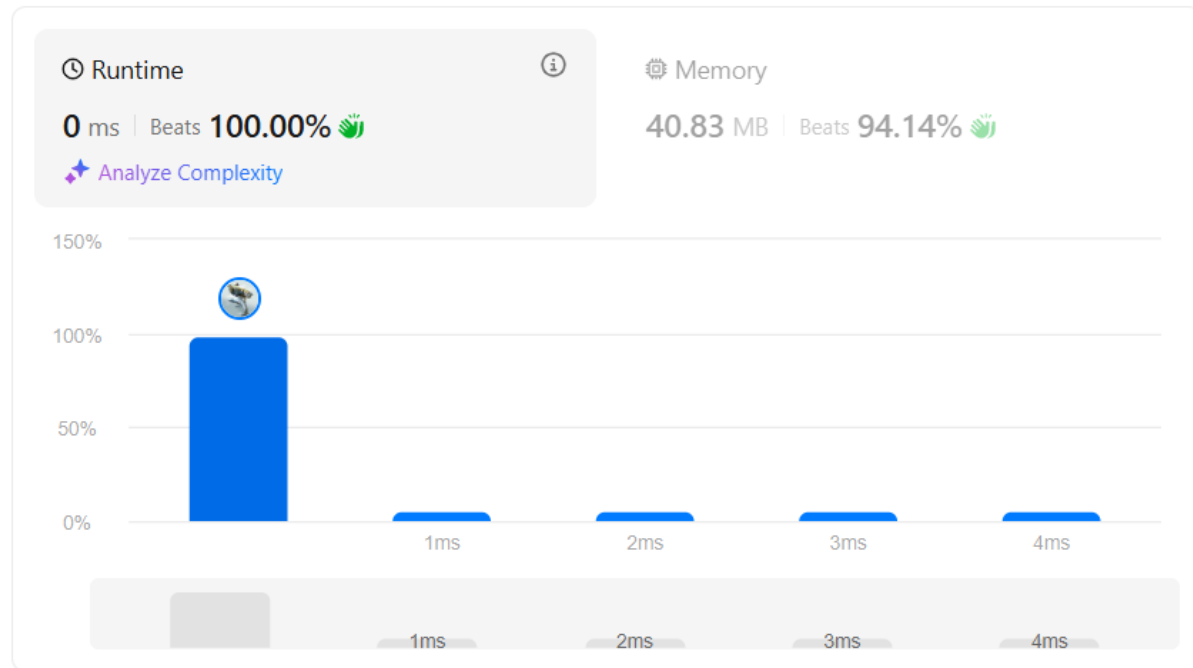
}

Accepted 36 / 36 testcases passed

 **Swatantra Yadav** submitted at Feb 16, 2025 20:06

 **Editorial**

 **Solution**



Merge two sorted linked lists


```
class Solution {  
    public ListNode mergeTwoLists(ListNode list1, ListNode list2) {  
  
        if(list1!=null && list2!=null){  
            if(list1.val<list2.val){  
                list1.next=mergeTwoLists(list1.next,list2);  
                return list1;  
            }  
        }  
    }  
}
```


```


        else{
            list2.next=mergeTwoLists(list1,list2.next);
            return list2;
        }
    }
    if(list1==null)
        return list2;
    return list1;
}
}

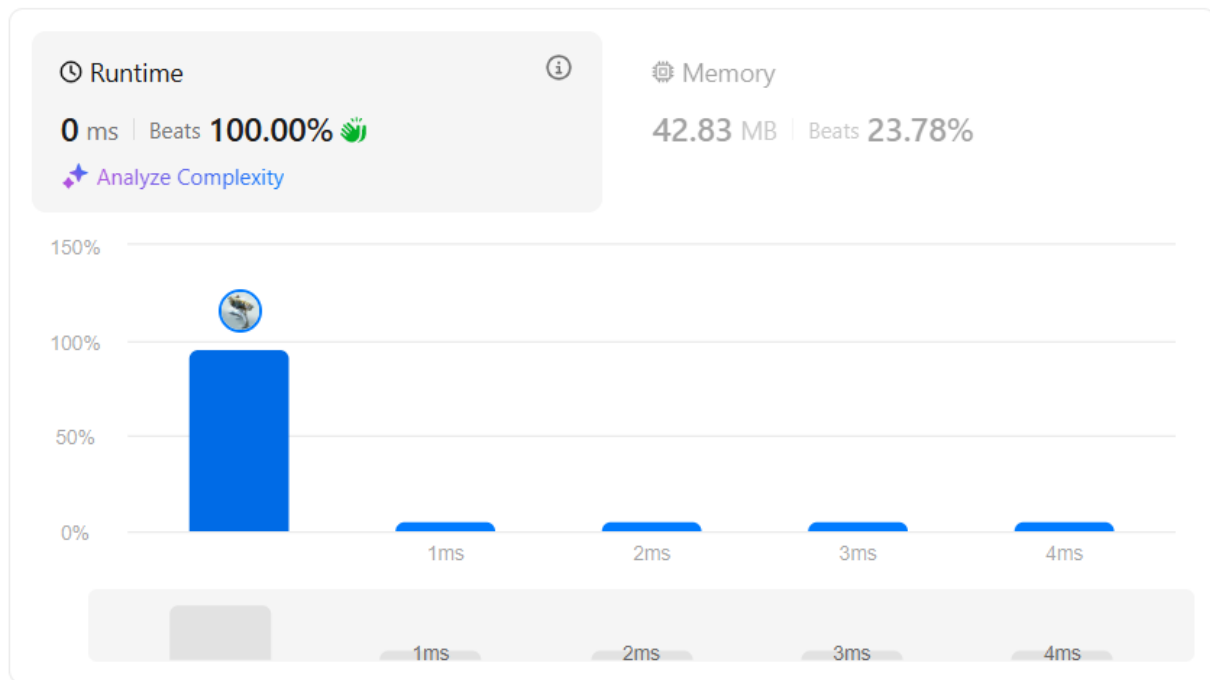
```

Accepted 208 / 208 testcases passed

 **Swatantra Yadav** submitted at Feb 16, 2025 20:08

 Editorial

 Solution



Remove duplicates from sorted lists 2

```

class Solution {
    public ListNode deleteDuplicates(ListNode head) {
        if (head == null || head.next == null) return head;

```

```
ListNode res = new ListNode(0,head);
ListNode prev = res;

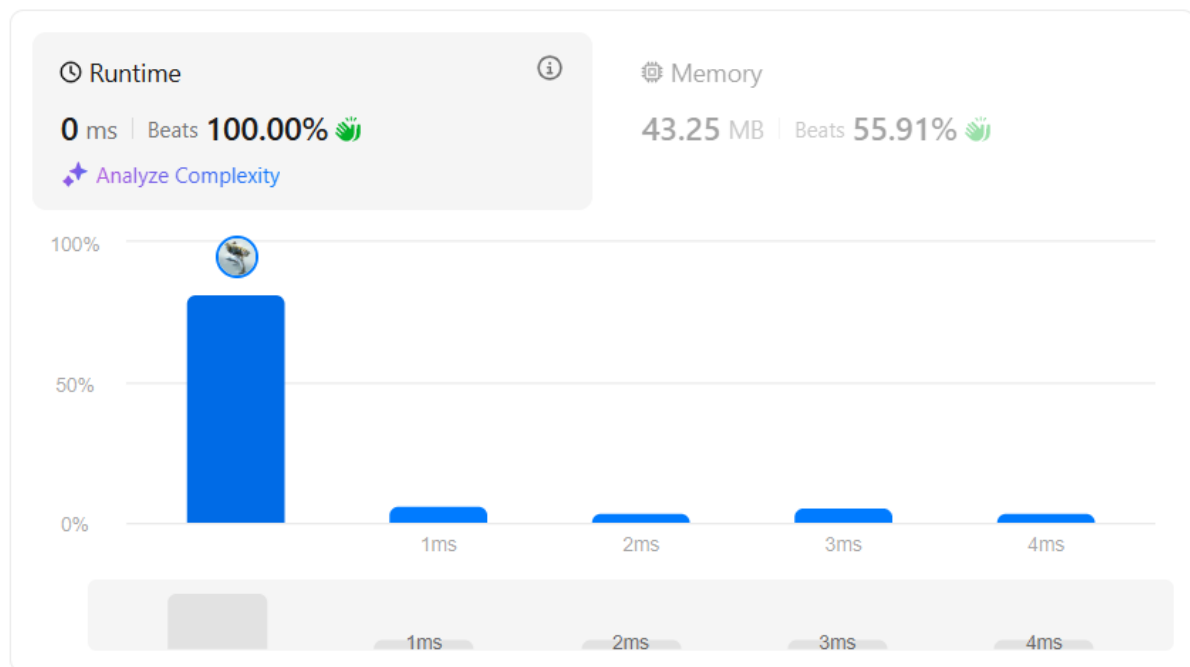
while(head != null && head.next != null){
    if(head.next.val == head.val){
        while(head.next != null && head.next.val == head.val){
            head = head.next;
        }
        prev.next = head.next;
    }else{
        prev = prev.next;
    }
    head = head.next;
}
return res.next;
}
}
```

Accepted 166 / 166 testcases passed

Swatantra Yadav submitted at Feb 16, 2025 20:11

Editorial

Solution



Detect a cycle in a linked list

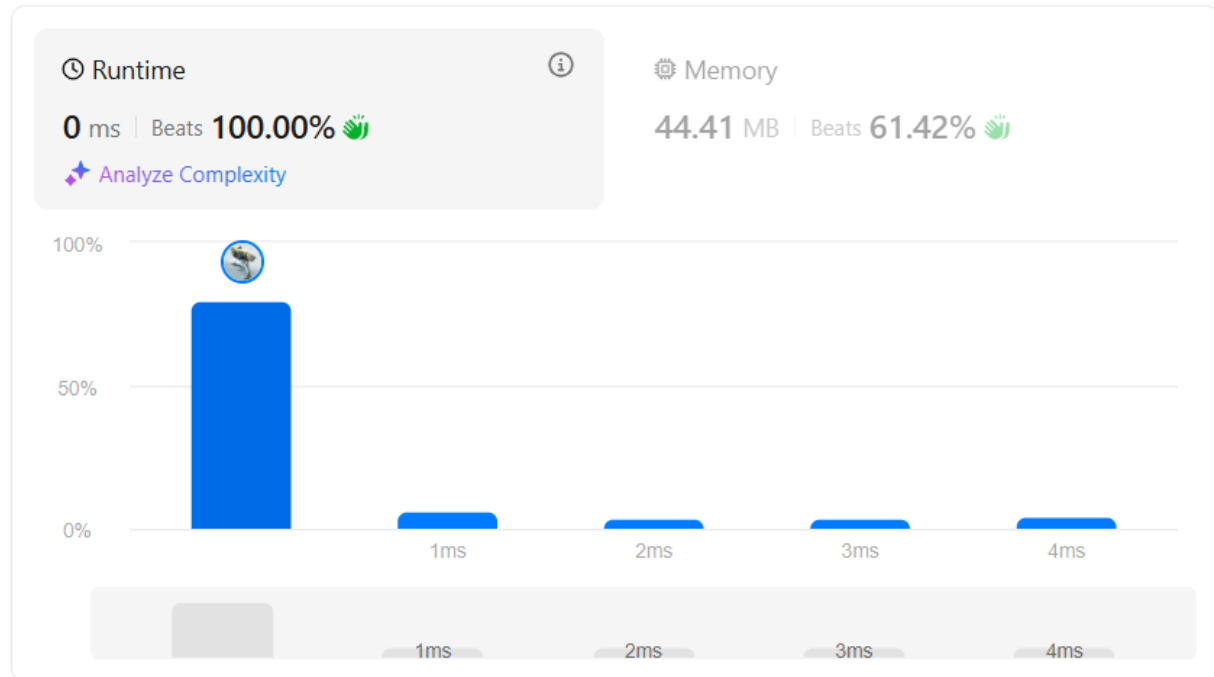
```
public class Solution {  
    public boolean hasCycle(ListNode head) {  
        ListNode slow = head, fast = head;  
        while (fast != null && fast.next != null) {  
            slow = slow.next;  
            fast = fast.next.next;  
            if (slow == fast) return true;  
        }  
        return false;  
    }  
}
```


Accepted 29 / 29 testcases passed

Swatantra Yadav submitted at Feb 16, 2025 20:27

Editorial

Solution



Reverse linked list 2

```
class Solution {  
    public ListNode reverseBetween(ListNode head, int left, int right) {  
        ArrayList<Integer> list = new ArrayList<>();  
  
        while(head != null){  
            list.add(head.val);  
            head = head.next;  
        }  
  
        int i = left-1, j = right-1;  
        while(i < j){  
            int temp = list.get(i);  
            list.set(i, list.get(j));  
            list.set(j, temp);  
        }  
    }  
}
```

```

        i++;
        j--;
    }


    ListNode dummy = new ListNode(0);
    ListNode curr = dummy;


    for(int ind = 0; ind < list.size(); ind++){
        curr.next = new ListNode(list.get(ind));
        curr = curr.next;
    }
    return dummy.next;
}
}

```

Accepted 44 / 44 testcases passed

 Swatantra Yadav submitted at Feb 16, 2025 20:33

 Editorial

 Solution

 Runtime

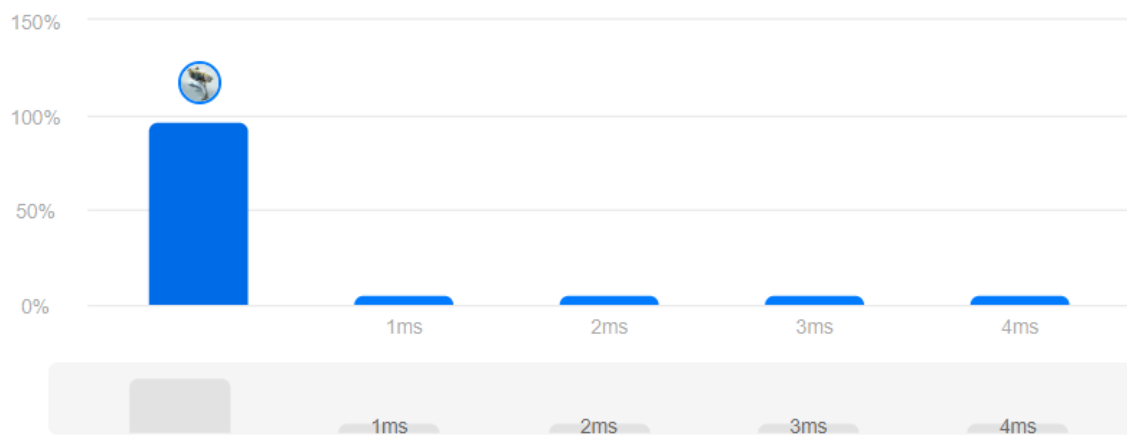


0 ms | Beats 100.00% 

 Analyze Complexity

 Memory

41.49 MB | Beats 39.63%





rotate a list


```
class Solution {  
    public ListNode rotateRight(ListNode head, int k) {  
        if (head == null || head.next == null || k == 0) {  
            return head;  
        }  
  
        int length = 1;  
        ListNode temp = head;  
  
        while (temp.next != null) {  
            temp = temp.next;  
            length++;  
        }  
  
        temp.next = head;  
        k = k % length;  
        k = length - k;  
  
        while (k-- > 0) {  
            temp = temp.next;  
        }  
  
        head = temp.next;  
        temp.next = null;  
  
        return head;  
    }  
}
```

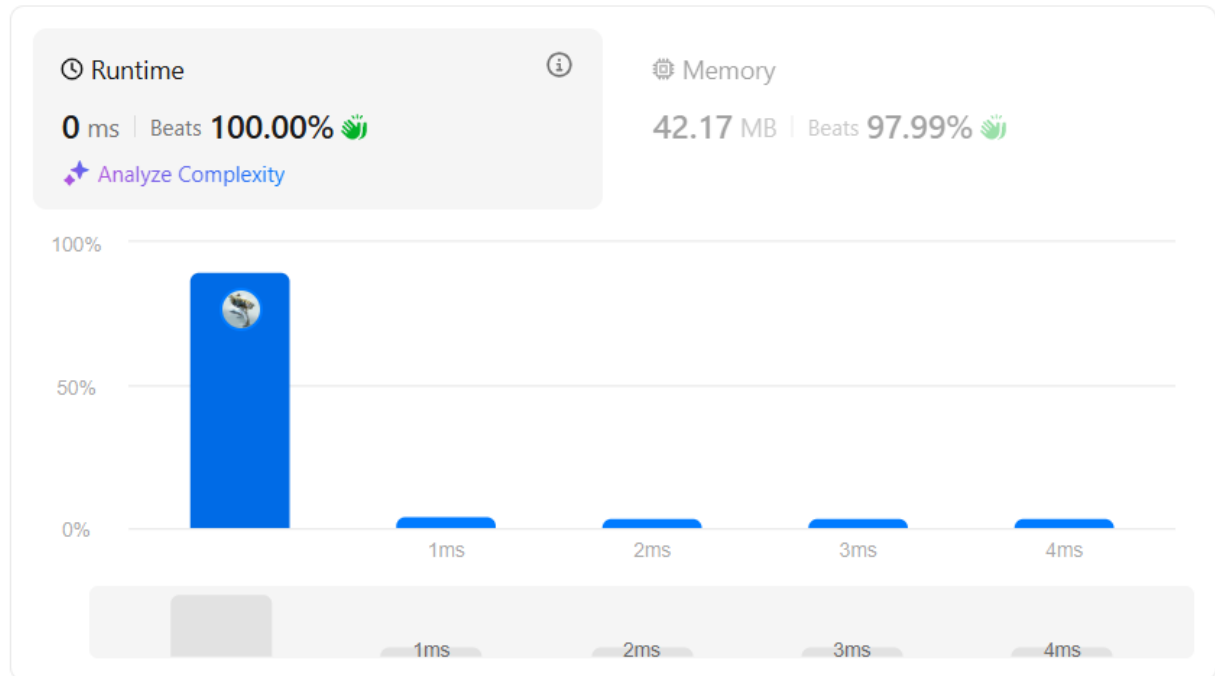
}

Accepted 232 / 232 testcases passed

 **Swatantra Yadav** submitted at Feb 16, 2025 20:35

 Editorial

 Solution





Detect a cycle in a linked list 2

```
public class Solution {  
    public ListNode detectCycle(ListNode head) {  
        ListNode slow = head, fast = head;  
        while (fast != null && fast.next != null) {  
            slow = slow.next;  
            fast = fast.next.next;  
            if (slow == fast) break;  
        }  
        if (fast == null || fast.next == null) return null;  
        while (head != slow) {  
            head = head.next;  
            slow = slow.next;  
        }  
    }  
}
```

```
}  
return head;  
}  
}
```

Accepted 18 / 18 testcases passed

 Swatantra Yadav submitted at Feb 16, 2025 20:36


 Editorial

 Solution

 Runtime



0 ms | Beats 100.00% 

 [Analyze Complexity](#)

 Memory

44.28 MB | Beats 96.39% 