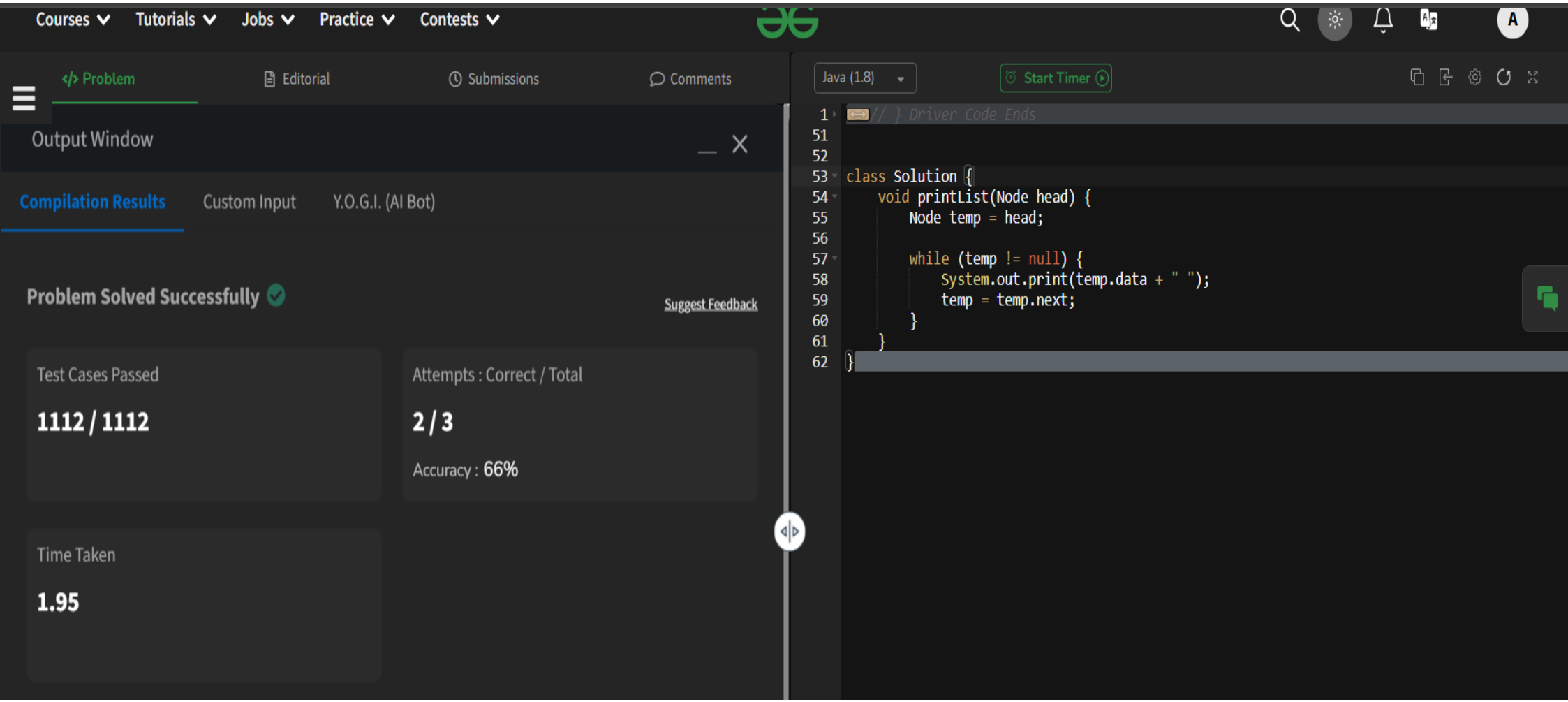

Courses
Tutorials
Jobs
Practice
Contests
A
Problem
Editorial
Submissions
Comments
Output Window
Compilation Results
Custom Input
Y.O.G.I. (AI Bot)
Problem Solved Successfully
Suggest Feedback
Test Cases Passed
1112 / 1112
Attempts : Correct / Total
2 / 3
Accuracy : 66%
Time Taken
1.95
Java (1.8)
Start Timer
// } Driver Code Ends
class Solution {
void printList(Node head) {
Node temp = head;
while (temp != null) {
System.out.print(temp.data + " ");
temp = temp.next;
}
}
}

Problem List | Run | Submit | 16 | Invite | Premium

Description | Accepted | Editorial | Submissions | Solutions

All Submissions

Accepted 168 / 168 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:46

Editorial | Solution

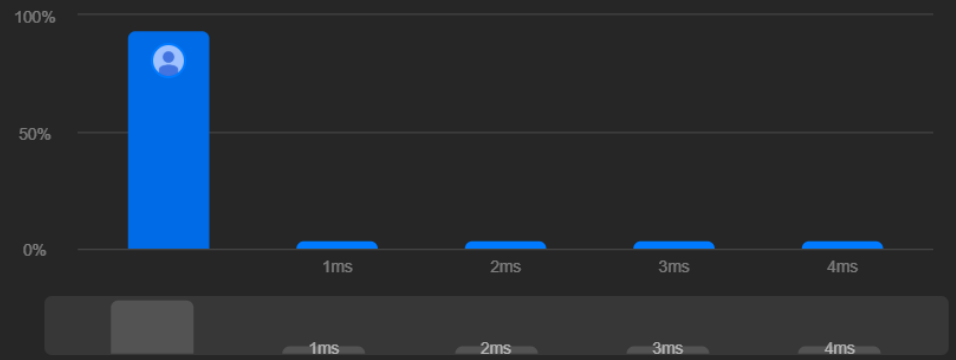
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.25 MB | Beats 41.51%



Code | Java

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

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

View more

Code

Java | Auto

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

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

        return res;
    }
}
```

Saved

Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | +

head =

[1,1,2]

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Accepted | Editorial | Submissions | Solutions

All Submissions

Accepted 28 / 28 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:48

Editorial | Solution

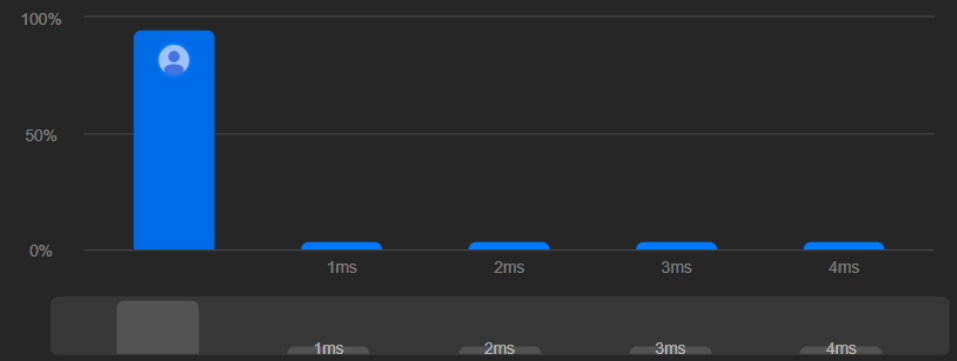
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.26 MB | Beats 91.58%



Code | Java

```java
class Solution {
    public ListNode reverseList(ListNode head) {

        ListNode prev = null;
        ListNode next = null;
        ListNode curr = head;

        while(curr != null)
```

View more

Code

Java | Auto

```java
class Solution {
    public ListNode reverseList(ListNode head) {

        ListNode prev = null;
        ListNode next = null;
        ListNode curr = head;

        while(curr != null)
        {
            next = curr.next;

            curr.next = prev;

            prev = curr;
            curr = next;
        }
        return prev;
    }
}
```

Saved | Ln 16, Col 10

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

head =

[1,2,3,4,5]

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Editorial | Submissions | Accepted | Solutions

All Submissions

Accepted 70 / 70 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:49

Editorial | Solution

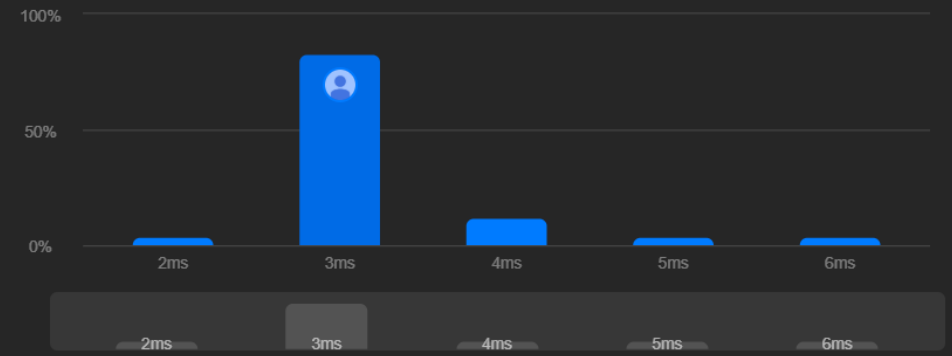
Runtime

3 ms | Beats 99.71%

Analyze Complexity

Memory

63.12 MB | Beats 77.77%



Code | Java

```
class Solution {
    public ListNode deleteMiddle(ListNode head) {
        if(head.next == null){
            return null;
        }
        ListNode slow = head;
        ListNode fast = head.next;
```

View more

Code

Java | Auto

```
1
2  class Solution {
3      public ListNode deleteMiddle(ListNode head) {
4          if(head.next == null){
5              return null;
6          }
7          ListNode slow = head;
8          ListNode fast = head.next;
9          while(fast.next != null && fast.next.next != null){
10             fast = fast.next.next;
11             slow = slow.next;
12         }
13         slow.next = slow.next.next;
14         return head;
15     }
16 }
```

Saved

Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

head =

[1,3,4,7,1,2,6]

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Accepted | Editorial | Submissions | Solutions

All Submissions

Accepted 208 / 208 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:50

Editorial | Solution

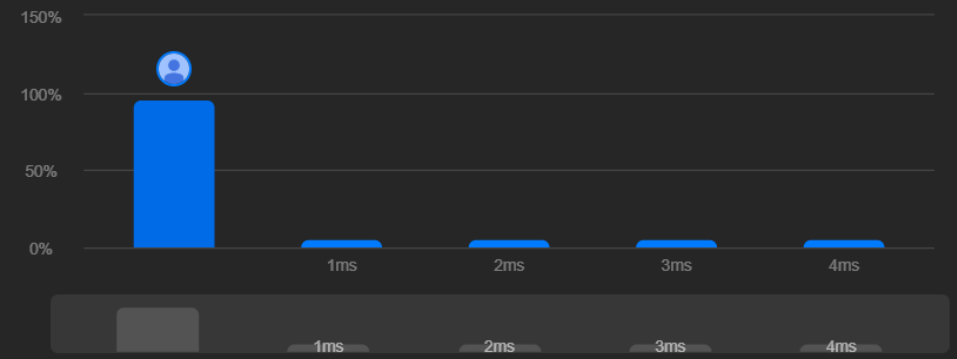
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.62 MB | Beats 44.78%



Code | Java

```
class Solution {
    public ListNode mergeTwoLists(ListNode list1, ListNode list2) {
        ListNode dummy = new ListNode();
        ListNode cur = dummy;

        while (list1 != null && list2 != null) {
            if (list1.val > list2.val) {
                cur.next = list2;
```

View more

Code

Java | Auto

```
1  class Solution {
2      public ListNode mergeTwoLists(ListNode list1, ListNode list2) {
3          ListNode dummy = new ListNode();
4          ListNode cur = dummy;
5
6          while (list1 != null && list2 != null) {
7              if (list1.val > list2.val) {
8                  cur.next = list2;
9                  list2 = list2.next;
10             } else {
11                 cur.next = list1;
12                 list1 = list1.next;
13             }
14             cur = cur.next;
15         }
16
17         cur.next = (list1 != null) ? list1 : list2;
18
19         return dummy.next;
20     }
21 }
```

Saved

Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

list1 =

[1,2,4]

list2 =

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Accepted | Editorial | Submissions | Solutions

All Submissions

Accepted 29 / 29 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:52

Editorial | Solution

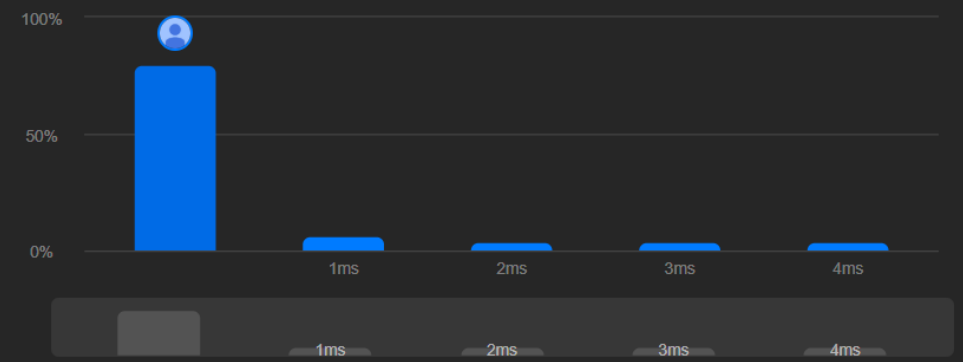
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

45.09 MB | Beats 7.17%



Code | Java

```java
public class Solution {
    public boolean hasCycle(ListNode head) {
        ListNode fast = head;
        ListNode slow = head;

        while (fast != null && fast.next != null) {
            fast = fast.next.next;
            slow = slow.next;
```

View more

Code

Java | Auto

```java
public class Solution {
    public boolean hasCycle(ListNode head) {
        ListNode fast = head;
        ListNode slow = head;

        while (fast != null && fast.next != null) {
            fast = fast.next.next;
            slow = slow.next;

            if (fast == slow) {
                return true;
            }
        }

        return false;
    }
}
```

Saved

Ln 17, Col 2

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

head =

[3,2,0,-4]

pos =

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Editorial | Submissions | Solutions | Accepted

All Submissions

Accepted 232 / 232 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:53

Editorial | Solution

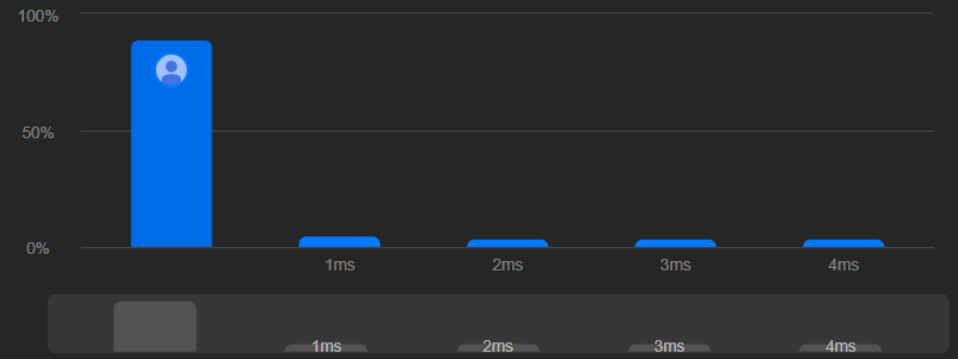
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.56 MB | Beats 62.47%



Code | Java

```java
class Solution {
    public ListNode rotateRight(ListNode head, int k) {
        // Edge case: If the list is empty or has only one node
        if(head == null || head.next == null) return head;

        // Step 1: Find the length of the linked list
        ListNode temp = head;
        int length = 0;
```

View more

Code

Java | Auto

```java
class Solution {
    public ListNode rotateRight(ListNode head, int k) {
        if(head == null || head.next == null) return head;
        ListNode temp = head;
        int length = 0;
        while(temp != null) {
            length++;
            temp = temp.next;
        }
        int rotation = k % length;
        for(int i = 0; i < rotation; i++) {
            ListNode last = head, prev = null;
            while(last.next != null) {
                prev = last;
                last = last.next;
            }
            last.next = head;
            prev.next = null;
            head = last;
        }

        return head;
    }
```

Saved

Ln 10, Col 35

Testcase | Test Result

Case 1 | Case 2 | +

head =

[1,2,3,4,5]

k =

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Editorial | Submissions | Solutions | Accepted

All Submissions

Accepted 30 / 30 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:55

Editorial | Solution

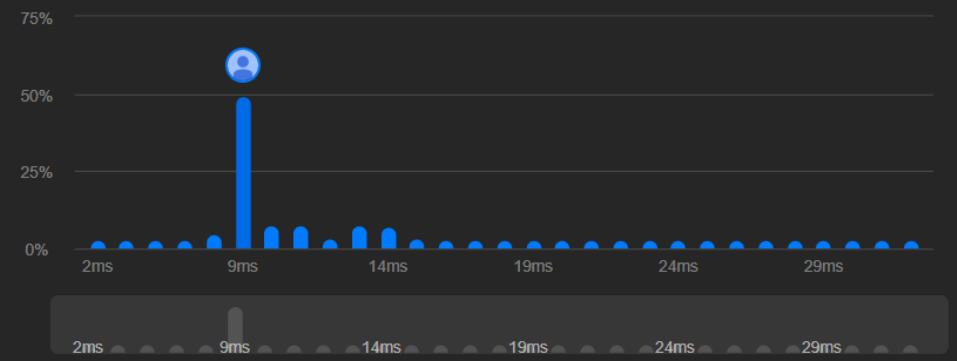
Runtime

9 ms | Beats **94.53%**

Analyze Complexity

Memory

57.14 MB | Beats **29.77%**



Code | Java

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

View more

Code

Java | Auto

```java
class Solution {
    public ListNode sortList(ListNode head) {
        if (head == null || head.next == null) return head;
        ListNode slow = head, fast = head.next;
        while (fast != null && fast.next != null) {
            slow = slow.next;
            fast = fast.next.next;
        }
        ListNode mid = slow.next;
        slow.next = null;
        ListNode left = sortList(head);
        ListNode right = sortList(mid);
        return merge(left, right);
    }
    private ListNode merge(ListNode l1, ListNode l2) {
        ListNode dummy = new ListNode(0);
        ListNode tail = dummy;

        while (l1 != null && l2 != null) {
            if (l1.val < l2.val) {
                tail.next = l1;
                l1 = l1.next;
            } else {
```

Saved

Ln 14, Col 6

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

head =

```
[4,2,1,3]
```

Source

Problem List | Run | Submit | 16 | Invite | Premium

Description | Accepted | Editorial | Submissions | Solutions

All Submissions

Accepted 134 / 134 testcases passed

Aarshdeep singh submitted at Mar 06, 2025 20:56

Editorial | Solution

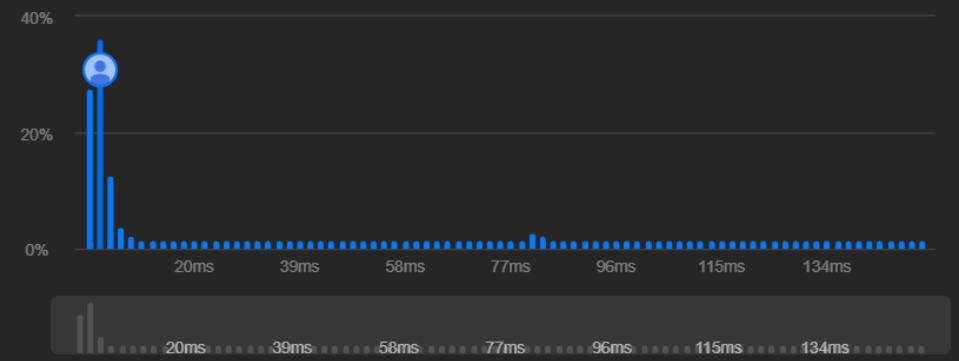
Runtime

2 ms | Beats 84.11%

Analyze Complexity

Memory

44.29 MB | Beats 89.43%



Code | Java

```java
import java.util.List;
class Solution {
    public ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        if (l1 == null) return l2;
        if (l2 == null) return l1;

        if (l1.val < l2.val) {
            l1.next = mergeTwoLists(l1.next, l2);
```

View more

Code

Java | Auto

```java
import java.util.List;
class Solution {
    public ListNode mergeTwoLists(ListNode l1, ListNode l2) {
        if (l1 == null) return l2;
        if (l2 == null) return l1;

        if (l1.val < l2.val) {
            l1.next = mergeTwoLists(l1.next, l2);
            return l1;
        } else {
            l2.next = mergeTwoLists(l1, l2.next);
            return l2;
        }
    }

    public ListNode mergeKLists(ListNode[] lists) {
        if (lists.length == 0) return null;
        return divideAndConquer(lists, 0, lists.length - 1);
    }

    private ListNode divideAndConquer(ListNode[] lists, int left, int right) {
        if (left == right) return lists[left];
```

Saved

Ln 1, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3 +

lists =

```
[[1,4,5],[1,3,4],[2,6]]
```

Source