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
1 / 1
Accuracy : 100%

Points Scored
1 / 1
Your Total Score: 7

Time Taken
1.82

Solve Next

Count Linked List Nodes | Delete Alternate Nodes | Insert in Middle of Linked List

Java (1.8) Start Timer

```
// } Driver Code Ends


/* Node is defined as
class Node {
    int data;
    Node next;
    Node(int x) {
        data = x;
        next = null;
    }
}*/
/*
    Print elements of a linked list on console
    Head pointer input could be NULL as well for empty list
*/

class Solution {
    // Function to display the elements of a linked list in the same line
    void printList(Node head) {
        Node temp = head;
        while (temp != null) {
            System.out.print(temp.data + " ");
            temp = temp.next;
        }
    }
}

```

Custom Input Compile & Run Submit

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 168 / 168 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:01

Editorial | Solution

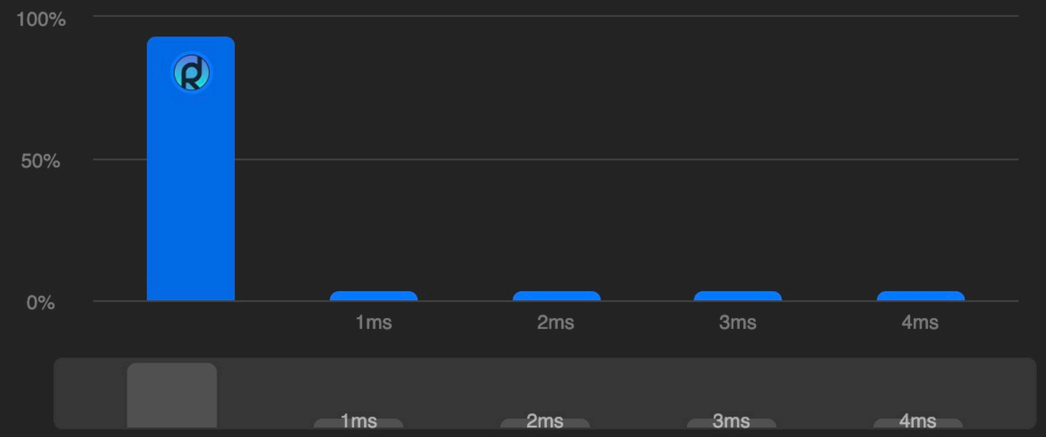
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.09 MB | Beats 72.19%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
7   *     ListNode(int val) { this.val = val; }
8   *     ListNode(int val, ListNode next) { this.val = val; this.next = next; }
9   * }
10  */
11  class Solution {
12      public ListNode deleteDuplicates(ListNode head) {
13          ListNode current = head;
14          while (current != null && current.next != null) {
15              if (current.val == current.next.val) {
16                  current.next = current.next.next; // Skip duplicate node
17              } else {
18                  current = current.next; // Move to next node
```

Saved Ln 24, Col 1

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2

Input

head =

[1,1,2]

Output

[1,2]

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 28 / 28 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:02

Editorial | Solution

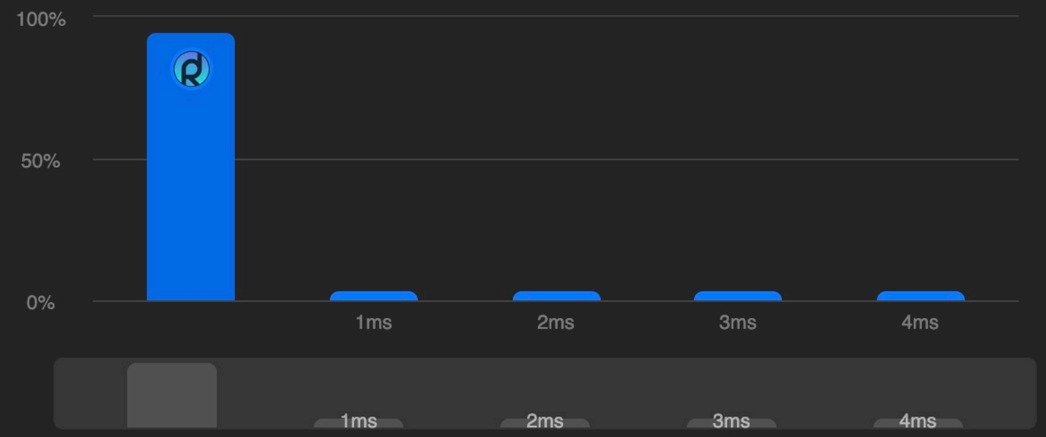
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.54 MB | Beats 50.77%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
16          while (current != null) {
17              ListNode nextNode = current.next; // Store next node
18              current.next = prev; // Reverse link
19              prev = current; // Move prev forward
20              current = nextNode; // Move current forward
21          }
22
23          return prev; // New head of the reversed list
24      }
25  }
26
```

Saved | Ln 26, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

[1,2,3,4,5]

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 70 / 70 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:03

Editorial | Solution

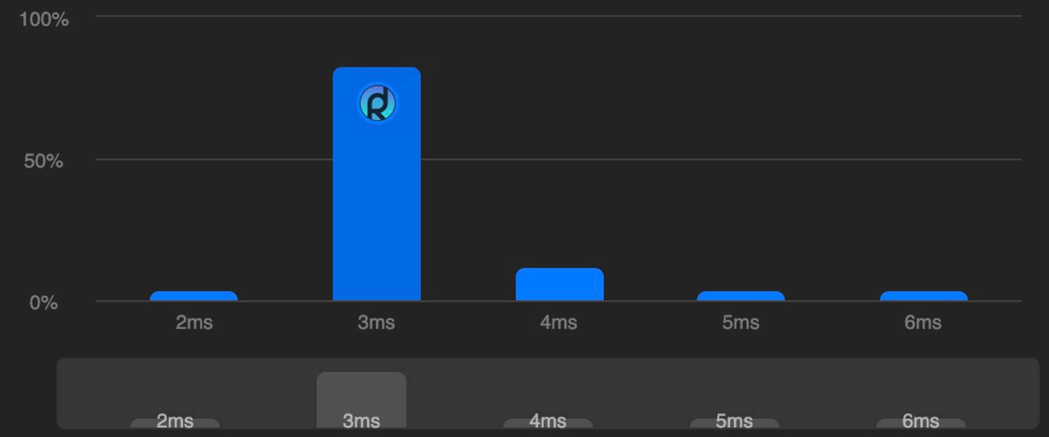
Runtime

3 ms | Beats 99.71%

Analyze Complexity

Memory

62.94 MB | Beats 86.91%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
17          while (fast != null && fast.next != null) {
18              prev = slow; // Store previous node of slow
19              slow = slow.next; // Move slow one step
20              fast = fast.next.next; // Move fast two steps
21          }
22
23          prev.next = slow.next; // Delete middle node
24          return head;
25      }
26  }
27
```

Saved | Ln 27, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

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

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 208 / 208 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:05

Editorial | Solution

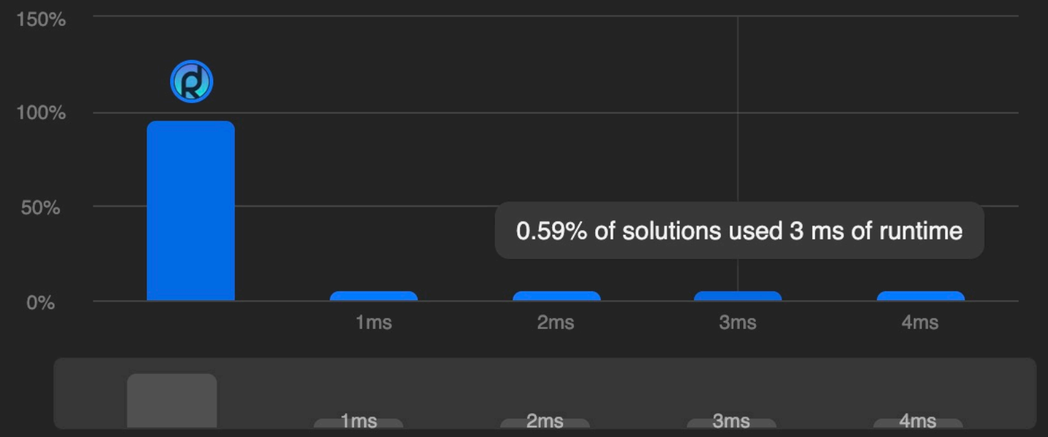
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.52 MB | Beats 58.03%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
                list2 = list2.next;
            }
            current = current.next;
        }

        current.next = (list1 != null) ? list1 : list2; // Attach the remaining nodes

        return dummy.next; // Return the merged list (excluding dummy node)
    }
}
```

Saved | Ln 32, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

list1 =

[1,2,4]

list2 =

[1,3,4]

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 29 / 29 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:06

Editorial | Solution

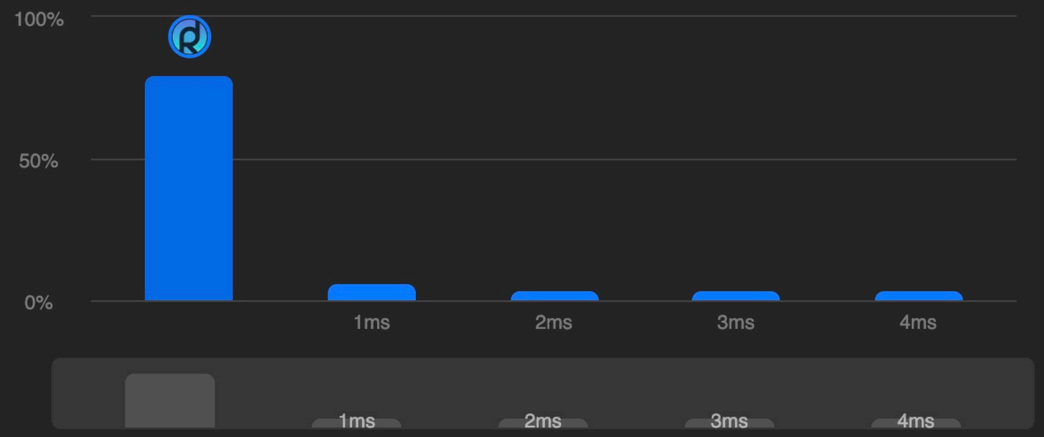
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.44 MB | Beats 61.48%



Code | Java

```
/**
 * Definition for singly-linked list.
 * class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode(int x) {
```

Code

Java | Auto

```
18          while (fast != null && fast.next != null) {
19              slow = slow.next;          // Move slow pointer one step
20              fast = fast.next.next;     // Move fast pointer two steps
21
22              if (slow == fast) return true; // Cycle detected
23          }
24
25          return false; // No cycle found
26      }
27  }
28
```

Saved | Ln 28, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

[3,2,0,-4]

pos =

1

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 232 / 232 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:07

Editorial | Solution

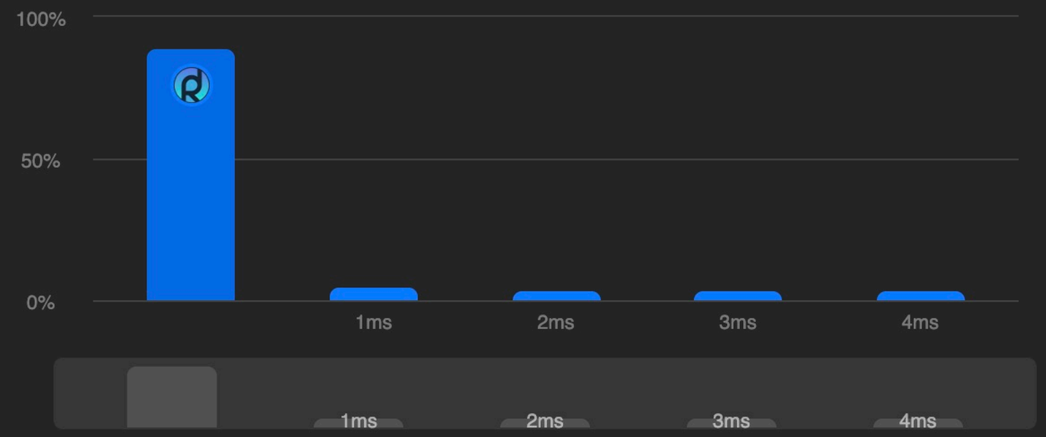
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.30 MB | Beats 89.63%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
31            newTail = newTail.next;
32        }
33
34        // Step 4: Break the circular list and set new head
35        head = newTail.next;
36        newTail.next = null;
37
38        return head;
39    }
40 }
41
```

Saved | Ln 41, Col 1

Testcase | Test Result

Case 1 | Case 2

head =

[1,2,3,4,5]

k =

2

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 30 / 30 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:08

Editorial | Solution

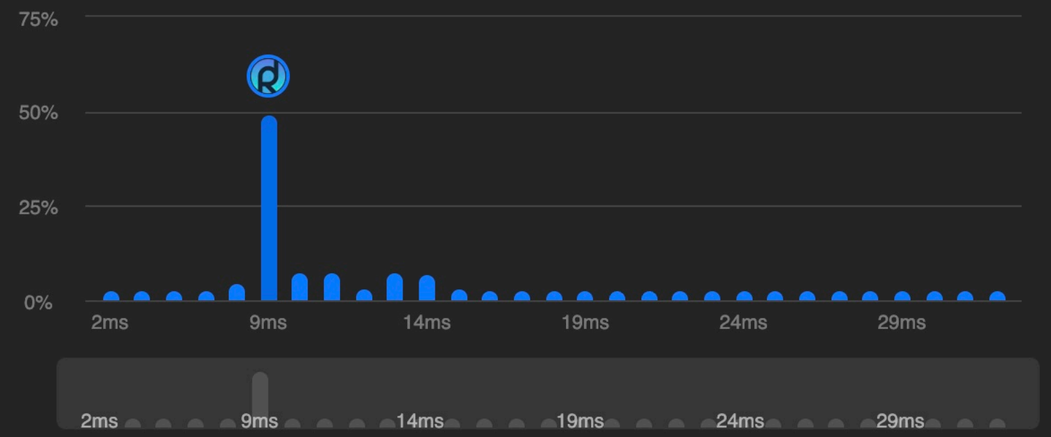
Runtime

9 ms | Beats 94.53%

Analyze Complexity

Memory

57.06 MB | Beats 37.55%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
46                  current.next = l2;
47                  l2 = l2.next;
48              }
49              current = current.next;
50          }
51
52          current.next = (l1 != null) ? l1 : l2;
53          return dummy.next;
54      }
55  }
56
```

Saved | Ln 56, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

head =

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

Source

Description | Accepted | Editorial | Solutions | Submissions

All Submissions

Accepted 134 / 134 testcases passed

jpAYB9fyeP submitted at Mar 06, 2025 22:09

Editorial | Solution

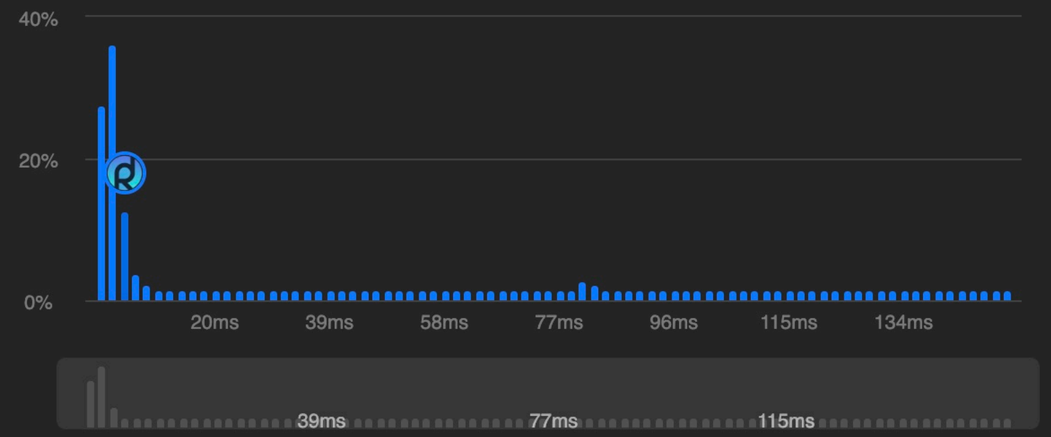
Runtime

4 ms | Beats 71.00%

Analyze Complexity

Memory

44.58 MB | Beats 48.85%



Code | Java

```
/**
 * Definition for singly-linked list.
 * public class ListNode {
 *     int val;
 *     ListNode next;
 *     ListNode() {}
```

Code

Java | Auto

```
34              current.next = smallest;
35              current = current.next;
36              if (smallest.next != null) {
37                  minHeap.offer(smallest.next);
38              }
39          }
40
41          return dummy.next;
42      }
43  }
44
```

Saved | Ln 44, Col 1

Testcase | Test Result

Case 1 | Case 2 | Case 3

lists =

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

Source