

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

3 / 3

Accuracy : 100%

Time Taken

1.8

You get marks only for the first correct submission if you solve the problem without viewing the full solution.

```
1 // } Driver Code Ends
51
52
53 /* Node is defined as
54 class Node {
55     int data;
56     Node next;
57     Node(int x) {
58         data = x;
59         next = null;
60     }
61 }*/
62 /*
63 Print elements of a linked list on console
64 Head pointer input could be NULL as well for empty list
65 */
66
67 class Solution {
68     // Function to display the elements of a linked list in same line
69     void printList(Node head) {
70         Node temp=head;
71         while(temp!=null)
72         {
73             System.out.print(temp.data+" ");
74             temp=temp.next;
75         }
76         // add code here.
77     }
78 }
```

← All Submissions

Accepted 158 / 168 testcases passed

Ragh... submitted at Feb 14, 2025 14:08

[Solution](#)

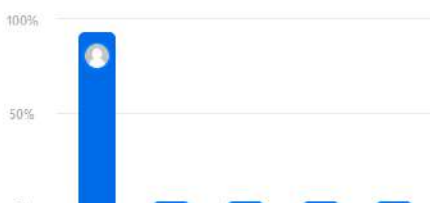
Runtime

0 ms | Beats 100.00%

[Analyze Complexity](#)

Memory

44.79 MB | Beats 6.66%



Java Auto

```
1 /**
2  * Definition for singly-linked list.
3  * public class ListNode {
4  *     int val;
5  *     ListNode next;
6  *     ListNode() {}
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
15         while (current != null && current.next != null) {
16             if (current.val == current.next.val) {
17                 current.next = current.next.next;
18             } else {
19                 current = current.next;
20             }
21         }
22     }
23 }
```

Saved

Ln 1, Col 1

☒ Testcase [Test Result](#)

Description Accepted Editorial Solutions

All Submissions

Accepted 28 / 28 testcases passed

Reg... submitted at Feb 14, 2025 14:09

Solution

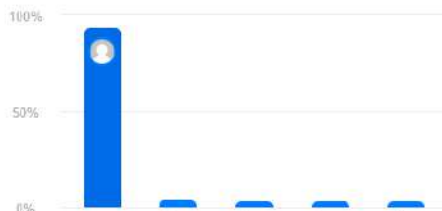
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.07 MB | Beats 98.14%



Code

Java Auto

```
5 * ListNode next;
6 * ListNode() {}
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 reverseList(ListNode head) {
13         ListNode node = null;
14
15         while (head != null) {
16             ListNode temp = head.next;
17             head.next = node;
18             node = head;
19             head = temp;
20         }
21
22         return node;
23     }
24 }
```

Saved

Testcase Test Result

Description Editorial Solutions Accepted

All Submissions

Accepted 70 / 70 testcases passed

Ragh... submitted at Feb 14, 2025 14:10

Solution

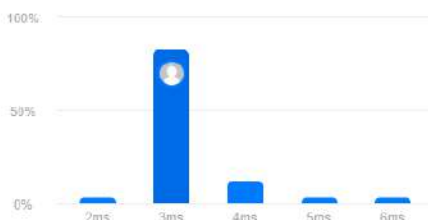
Runtime

3 ms | Beats 99.75%

Analyze Complexity

Memory

62.88 MB | Beats 91.91%



Code

Java Auto

```
6 * ListNode() {}
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 deleteMiddle(ListNode head) {
13         if(head == null) return null;
14         ListNode prev = new ListNode(0);
15         prev.next = head;
16         ListNode slow = prev;
17         ListNode fast = head;
18         while(fast != null && fast.next != null){
19             slow = slow.next;
20             fast = fast.next.next;
21         }
22         slow.next = slow.next.next;
23         return prev.next;
24     }
25 }
```

Saved

Ln 25, Col 2

Testcase Test Result

Accepted Runtime: 0 ms

← All Submissions

Accepted 208 / 208 testcases passed

Ragh... submitted at Feb 14, 2025 14:11

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

42.39 MB | Beats 85.87%

Java Auto

15

16

17

18

19

20

21

22

23

24

25

26

27

28

29

30

31

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

cur.next = (list1 != null) ? list1 : list2;

return dummy.next;
```

Saved

Testcase

Test Result

Description Editorial Solutions Accepted X

All Submissions

Accepted 166 / 166 testcases passed

Ragh... submitted at Feb 14, 2025 14:11



Solution

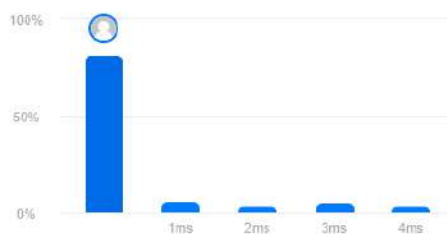
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

43.32 MB | Beats 44.70%



Code

Java Auto

```
18  ListNode curr = head;
19  ListNode temp = head.next;
20  boolean flag = false;
21
22  while(temp != null){
23      if(curr.val != temp.val){
24          if(flag == true){
25              prev.next = temp;
26              flag = false;
27          }
28          else{
29              prev = prev.next;
30          }
31      }
32      else if(curr.val == temp.val){
33          flag = true;
34      }
35      temp = temp.next;
36      curr = curr.next;
37  }
38  if(flag == true){
```

Saved

Ln 43

Testcase Test Result

Accepted Runtime: 0 ms

Accepted 29 / 29 testcases passed

Ragh... submitted at Feb 14, 2025 14:12



Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

44.61 MB | Beats 38.28%



```
8 *
9 *
10 * }
11 */
12 public class Solution {
13     public boolean hasCycle(ListNode head) {
14         ListNode fast = head;
15         ListNode slow = head;
16
17         while (fast != null && fast.next != null) {
18             fast = fast.next.next;
19             slow = slow.next;
20
21             if (fast == slow) {
22                 return true;
23             }
24         }
25
26         return false;
27     }
28 }
```

Saved

Testcase Test Result