

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 / 3

Accuracy : 33%

Points Scored

1 / 1

Your Total Score: 26

Time Taken

0.07

Solve Next

Count Linked List Nodes

Delete Alternate Nodes

Insert in Middle of Linked List

C++ (g++ 5.4)

Start Timer

1

19

20

21

22

23

24

25

26

27

28

29

30

31

32

33

34

35

36

37

38

39

40

41

42

43

44

45

46

47

48

49

50

// active code ends

/*

struct Node {

int data;

struct Node* next;

Node(int x) {

data = x;

next = nullptr;

}

};

*/

Print elements of a linked list on console

Head pointer input could be NULL as well for empty list

*/

class Solution {

public:

// function to display the elements of a linked list in same line

void printlist(Node *head) {

// your code goes here

Node *temp=head;

while(temp!=NULL)

{

cout<<temp->data<<" ";

temp=temp->next;

}

}

};

// active code ends

Description Editorial Solutions Accepted Submissions

All Submissions

Accepted 168 / 168 testcases passed

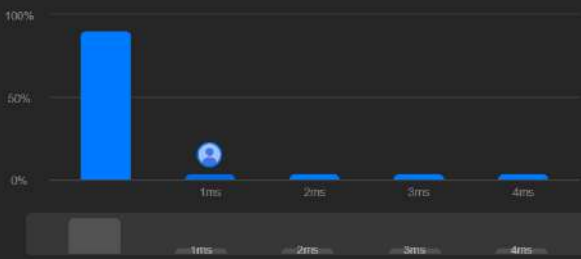
Muskan Sharma submitted at Mar 06, 2025 20:54

Editorial Solution

Runtime 1 ms | Beats 9.78%

Memory 16.29 MB | Beats 35.10%

Analyze Complexity



Code C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 */
```

Code

C++ Auto

```
16     ListNode* current = head;
17
18     while (current && current->next) {
19         if (current->val == current->next->val) { // Compare values
20             ListNode* temp = current->next;
21             current->next = current->next->next; // Remove duplicate node
22             delete temp; // Free memory
23         } else {
24             current = current->next; // Move to next distinct element
25         }
26     }
27 }
```

Saved

Ln 30, Col 4

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =

[1,1,2]

Output

[1,2]

Expected

DescriptionAccepted xEditorialSolutionsSubmissions

← All Submissions

Accepted 28 / 28 testcases passed

Muskan Sharma submitted at Mar 06, 2025 20:56

EditorialSolution

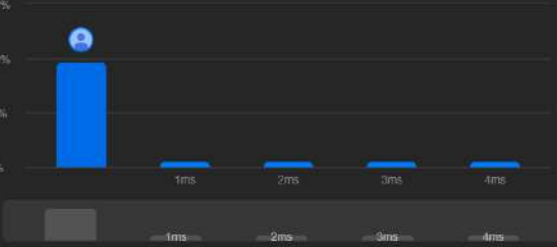
Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

13.28 MB | Beats 90.63%



Case	Runtime (ms)
Case 1	0
Case 2	~0.5
Case 3	~0.5
Case 4	~0.5
Case 5	~0.5

Code C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 */
```

Code

C++ vAuto

```
1 //
10 */
11 class Solution {
12 public:
13     ListNode* reverseList(ListNode* head) {
14         ListNode* prev = nullptr;
15         ListNode* current = head;
16
17         while (current) {
18             ListNode* nextNode = current->next; // Store the next node
19             current->next = prev; // Reverse the link
20             prev = current; // Move "prev" to current node
21         }
22     }
23 }
```

Saved

Ln 27, Col 1

TestcaseTest Result

Accepted Runtime: 0 ms

Case 1Case 2Case 3

Input

head =

[1,2,3,4,5]

Output

[5,4,3,2,1]

Expected

Problem List

RunSubmit

Premium

DescriptionAcceptedEditorialSolutionsSubmissions

All Submissions

Accepted70 / 70 testcases passed
Muskan Sharma submitted at Mar 06, 2025 21:03

Runtime
3 ms | Beats 45.30%
Analyze Complexity

Memory
312.04 MB | Beats 55.26%

Interval	Percentage
1ms	45.30%
2ms	~5%
3ms	~10%
4ms	~15%
5ms	~5%
6ms	~2%
7ms	~2%

Code

C++Auto

```
25 }
26
27 // Delete the middle node
28 prev->next = slow->next;
29 delete slow;
30
31 return head;
32 }
33 };
34
```

SavedLn 34, Col 1

TestcaseTest Result

AcceptedRuntime: 0 ms

Case 1Case 2Case 3

Input
head =
[1,3,4,7,1,2,6]

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

CodeC++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;

```

Problem List

Run

Submit

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted 208 / 208 testcases passed

Muskan Sharma submitted at Mar 06, 2025 21:06

Editorial

Solution


Runtime

0 ms Beats 100.00%

Analyze Complexity

Memory

19.42 MB Beats 62.23%



Runtime	Beats
0 ms	100.00%
1 ms	~0%
2 ms	~0%
3 ms	~0%
4 ms	~0%

Code

C++

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if (!list1) return list2; // If list1 is empty, return list2
        if (!list2) return list1; // If list2 is empty, return list1
        if (list1->val < list2->val) {
            list1->next = mergeTwoLists(list1->next, list2);
            return list1;
        } else {
            list2->next = mergeTwoLists(list1, list2->next);
            return list2;
        }
    }
};
```

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

list1 =

[1,2,4]

list2 =

[1,3,4]

Output

Problem List

Accepted Editorial Solutions Submissions

All Submissions

Accepted 29 / 29 testcases passed

Muskan Sharma submitted at Mar 06, 2025 21:09

Editorial Solution

Runtime

12 ms | Beats 40.41%

Analyze Complexity

Memory

11.88 MB | Beats 54.03%

Code

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =

[3, 2, 0, -4]

pos =

1

Output

Code

C++ Auto

```
16     slow = slow->next; // Moves 1 step
17     fast = fast->next->next; // Moves 2 steps
18
19     if (slow == fast) return true; // Cycle detected
20 }
21
22 return false; // No cycle found
23 }
24 };
25
```

Saved

Ln 25, Col 1

Testcase

Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =

[3, 2, 0, -4]

pos =

1

Output

Code

C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 */
```


Description | **Accepted** | Editorial | Solutions | Submissions

All Submissions

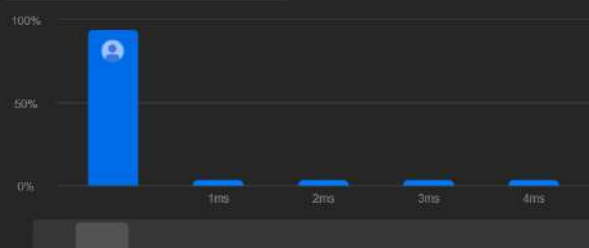
Accepted 232 / 232 testcases passed

Muskan Sharma submitted at Mar 06, 2025 21:13

Editorial Solution

Runtime 0 ms | Beats 100.00%  **Memory** 16.45 MB | Beats 31.91%

Analyze Complexity



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() {}
 *     ListNode(int x) { val = x; }
 *     ListNode(int x, ListNode *next) { val = x; next = next; }
 * };
 */
```

Code

C++ | Auto

```
33     }
34
35     // Step 4: Set the new head and break the cycle
36     head = newTail->next;
37     newTail->next = nullptr;
38
39     return head;
40 }
41 };
42
```

Saved

Ln 42, Col 1

Testcase | **Test Result**

Accepted Runtime: 0 ms

Case 1 Case 2

Input

head =
[1,2,3,4,5]

k =
2

Description | Accepted x | Editorial | Solutions | Submissions

← All Submissions

Accepted 30 / 30 testcases passed

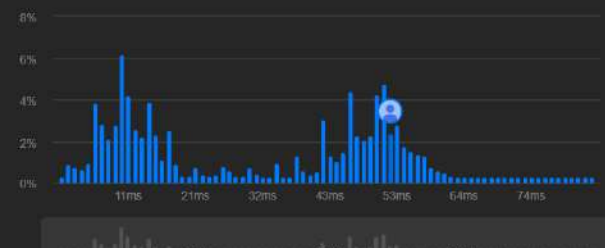
Muskan Sharma submitted at Mar 06, 2025 21:15

Editorial Solution

Runtime 52 ms | Beats 25.94%

Memory 75.92 MB | Beats 6.70%

Analyze Complexity



Code | C++

```
/*++
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() {}
 *     ListNode(int x) { val = x; }
 *     ListNode(int x, ListNode *next) { val = x; next = next; }
 * };
 */
```

Code

```
39 ListNode* getMiddle(ListNode* head) {
40     ListNode* slow = head;
41     ListNode* fast = head;
42     ListNode* prev = nullptr;
43
44     while (fast && fast->next) {
45         prev = slow;
46         slow = slow->next;
47         fast = fast->next->next;
48     }
49 }
```

Saved

Testcase Test Result

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

head =
[4,2,1,3]

Output

[1,2,3,4]

Problem List
Run
Submit

Description | Accepted x | Editorial | Solutions | Submissions

All Submissions

Accepted 134 / 134 testcases passed

Muskan Sharma submitted at Mar 06, 2025 21:16

Runtime
 0 ms | Beats 100.00%
[Analyze Complexity](#)

Memory
 18.28 MB | Beats 89.49%

Code

```
C++
// ...
if (smallest->next) {
    minHeap.push(smallest->next);
}
return dummy->next;
};
```

Saved

Testcase

Accepted Runtime: 0 ms

Case 1 Case 2 Case 3

Input

lists =

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

Output

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

Expected