



Problem

Editorial

Submissions

Comments

C++ (g++ 5.4)

Start Timer



Output Window



Compilation Results

Custom Input

Compilation Completed

Case 1

Input:

12

Your Output:

12

Expected Output:

12



```
1 // } Driver Code Ends
19
20 /*
21 struct Node {
22     int data;
23     struct Node* next;
24
25     Node(int x) {
26         data = x;
27         next = nullptr;
28     }
29 };
30 */
31 /*
32 Print elements of a linked list on console
33 Head pointer input could be NULL as well for empty list
34 */
35
36 class Solution {
37 public:
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
51
52
53 // } Driver Code Ends
```

Accepted 168 / 168 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:25

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.32 MB | Beats 11.13%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

Code

C++ Auto

```
10  /**
11  class Solution {
12  public:
13      ListNode* deleteDuplicates(ListNode* head) {
14          ListNode* current = head;
15
16          while (current != nullptr && current->next != nullptr) {
17              if (current->val == current->next->val) {
18
19                  ListNode* temp = current->next;
20                  current->next = current->next->next;
21                  delete temp;
22              } else {
23
24                  current = current->next;
25              }
26          }
27
28          return head;
29      }
```

Saved

Ln 21, Col 30

Testcase Test Result

Case 1 Case 2 +

head =

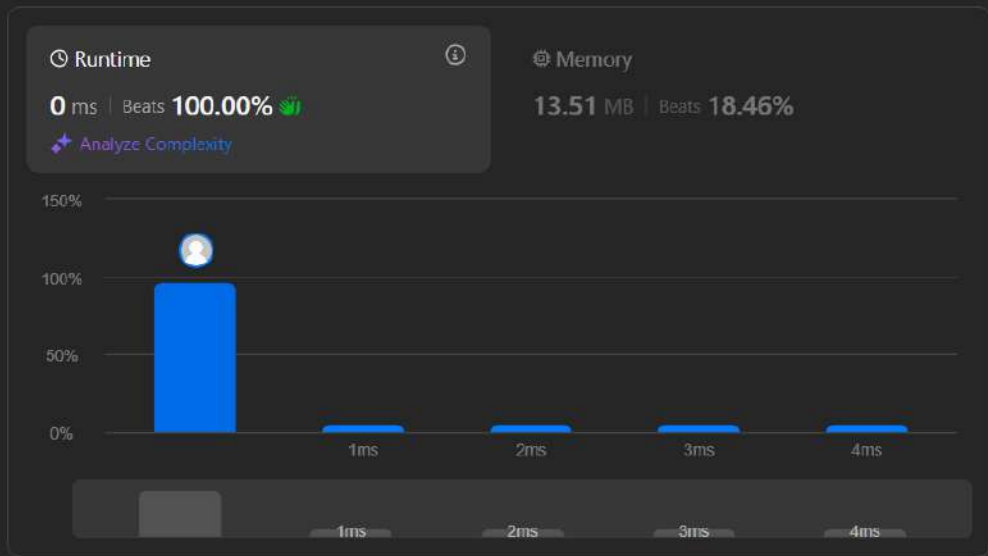
[1,1,2]

Source ?

Accepted 28 / 28 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:28

Editorial Solution



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * }
```

</> Code

C++ v Auto

```
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* reverseList(ListNode* head) {
14         ListNode* prev = nullptr;
15         ListNode* current = head;
16         ListNode* next = nullptr;
17
18         while (current != nullptr) {
19             next = current->next;
20             current->next = prev;
21             prev = current;
22             current = next;
23         }
24
25         return prev;
26     }
27 };
28
```

Saved

Ln 25, Col 23

☒ Testcase < > Test Result

Case 1 Case 2 Case 3 +

head =

[1,2,3,4,5]

</> Source ?

Accepted 70 / 70 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:30

Editorial

Solution

Runtime

2 ms | Beats 50.07%

Analyze Complexity

Memory

312.10 MB | Beats 17.89%



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

Code

C++ Auto

```
10  /*
11  class Solution {
12  public:
13      ListNode* deleteMiddle(ListNode* head) {
14          if (!head || !head->next) return nullptr;
15
16          ListNode* slow = head;
17          ListNode* fast = head;
18          ListNode* prev = nullptr;
19
20          while (fast && fast->next) {
21              prev = slow;
22              slow = slow->next;
23              fast = fast->next->next;
24          }
25
26          prev->next = slow->next;
27          delete slow;
28
29          return head;
30      }
31  };
32  */
```

Saved

Ln 32, Col 1

Testcase Test Result

Case 1

Case 2

Case 3

+

head =

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

Source

Accepted 208 / 208 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:34

Editorial Solution



Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

```
1 /**
2  * Definition for singly-linked list.
3  * struct ListNode {
4  *     int val;
5  *     ListNode *next;
6  *     ListNode() : val(0), next(nullptr) {}
7  *     ListNode(int x) : val(x), next(nullptr) {}
8  *     ListNode(int x, ListNode *next) : val(x), next(next) {}
9  * };
10 */
11 class Solution {
12 public:
13     ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
14         if (!list1) return list2;
15         if (!list2) return list1;
16
17         ListNode* dummy = new ListNode(-1);
18         ListNode* current = dummy;
19
20         while (list1 && list2) {
21             if (list1->val < list2->val) {
22                 current->next = list1;
23                 list1 = list1->next;
24             } else {
25                 current->next = list2;
26                 list2 = list2->next;
27             }
28             current = current->next;
29         }
30         if (list1) current->next = list1;
31         if (list2) current->next = list2;
32         return dummy->next;
33     }
34 };
```

Saved

Ln 39, Col 1

Case 1 Case 2 Case 3 +

list1 =

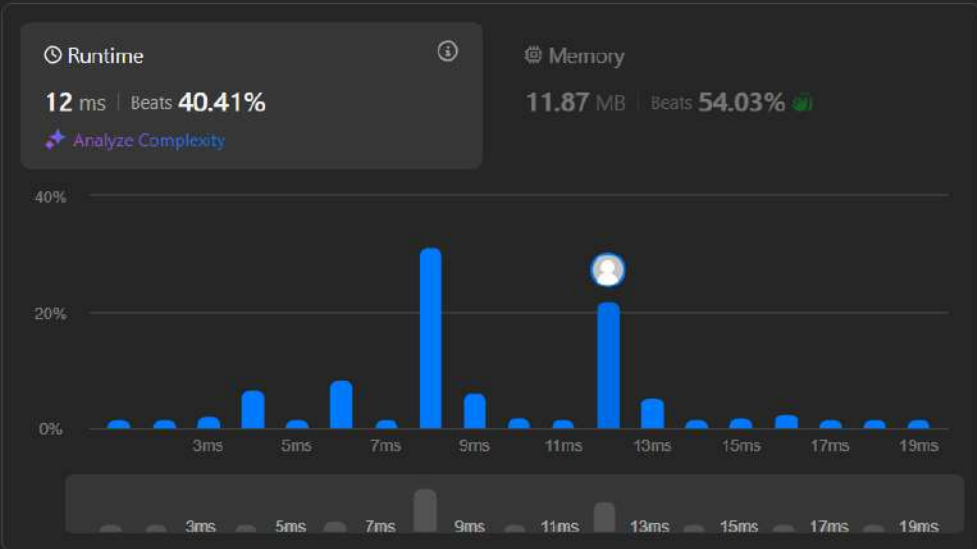
[1,2,4]

Source ⓘ

Accepted 29 / 29 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:35

Editorial Solution



Code | C++

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

```
7  *};
8  */
9  class Solution {
10 public:
11     bool hasCycle(ListNode* head) {
12         if (!head || !head->next) return false;
13
14         ListNode* slow = head;
15         ListNode* fast = head;
16
17         while (fast && fast->next) {
18             slow = slow->next;
19             fast = fast->next->next;
20             if (slow == fast) return true;
21         }
22
23         return false;
24     }
25 };
26
```

Case 1 Case 2 Case 3 +

head =

[3,2,0,-4]

</> Source ⓘ

Problem List

Run

Submit

1

Premium

Description

Accepted

Editorial

Solutions

Submissions

All Submissions

Accepted

232 / 232 testcases passed

Ashutosh2215 submitted at Mar 06, 2025 20:36

Editorial

Solution

Runtime

0 ms | Beats 100.00%

Analyze Complexity

Memory

16.51 MB | Beats 4.06%

Runtime (ms)	Beats (%)
0	100.00%
1	~0.01%
2	~0.01%
3	~0.01%
4	~0.01%

Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * }
```

Code

C++

```
public:
    ListNode* rotateRight(ListNode* head, int k) {
        if (!head || !head->next || k == 0) return head;

        ListNode* temp = head;
        int length = 1;

        while (temp->next) {
            temp = temp->next;
            length++;
        }

        temp->next = head;
        k = k % length;
        int stepsToNewHead = length - k;

        while (stepsToNewHead--) {
            temp = temp->next;
        }

        head = temp->next;
    }
```

Saved

Ln 38, Col 1

Testcase


Test Result





Case 1 Case 2 +




head =

[1,2,3,4,5]

Source

 Problem List < > 🔍

 Run  Submit  

  🔥 1  Premium

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

← All Submissions 🔗

Accepted 30 / 30 testcases passed


Ashutosh2215 submitted at Mar 06, 2025 20:37

Editorial

Solution

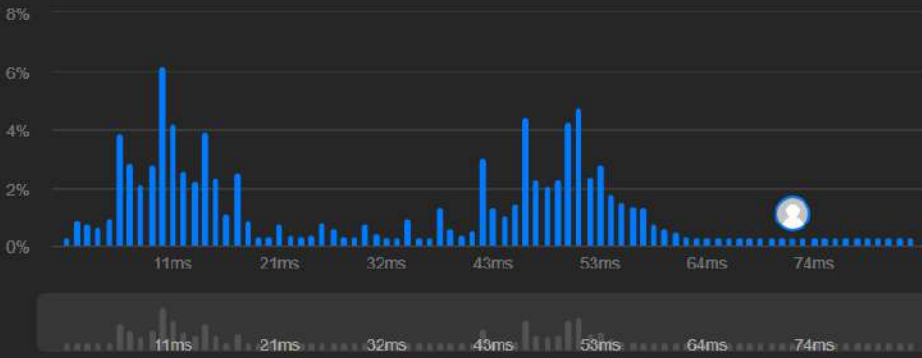
🕒 Runtime ⓘ

72 ms | Beats 6.71%

 Analyze Complexity

💾 Memory

86.44 MB | Beats 5.21%



The chart displays the performance of the solution across various test cases. The x-axis represents time in milliseconds (ms) from 0 to 74ms, and the y-axis represents the percentage of test cases passed from 0% to 8%. The blue bars show the distribution of test cases, with a peak around 11ms and another around 53ms. A white circle highlights the current solution's performance at 72ms.

Code | C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode() : val(0), next(nullptr) {}
 *     ListNode(int x) : val(x), next(nullptr) {}
 *     ListNode(int x, ListNode *next) : val(x), next(next) {}
 * };
 */
```

</> Code

C++ ▾ 🔒 Auto

⋮

🔖

⌗

↺

↻

↗

```
12 public:
13     ListNode* merge(ListNode* left, ListNode* right) {
14         if (!left) return right;
15         if (!right) return left;
16
17         ListNode* dummy = new ListNode(-1);
18         ListNode* current = dummy;
19
20         while (left && right) {
21             if (left->val <= right->val) {
22                 current->next = left;
23                 left = left->next;
24             } else {
25                 current->next = right;
26                 right = right->next;
27             }
28             current = current->next;
29         }
30
31         if (left) current->next = left;
32         if (right) current->next = right;
```

Saved

Ln 64, Col 1

☒ Testcase | > Test Result

Case 1 Case 2 Case 3 +

head =

[4,2,1,3]

</> Source ⓘ