

Merge two sorted 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) swap(l1, l2);

        ListNode* ans = l1;

        while (l1 != NULL && l2 != NULL) {
            ListNode* temp = NULL;
            while (l1 != NULL && l1->val <= l2->val) {
                temp = l1;
                l1 = l1->next;
            }
            temp->next = l2;

            swap(l1, l2);
        }
        return ans;
    }
};
```


Description | Accepted | Editorial | Solutions | Submissions

← All Submissions

Accepted 208 / 208 testcases passed
12ananya submitted at Feb 13, 2025 21:55

Runtime
0 ms | Beats 100.00%
Analyze Complexity

Memory
19.65 MB | Beats 10.67%



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

Code | C++

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        if (l1 == NULL) return l2; // If l1 is empty, return l2
        if (l2 == NULL) return l1; // If l2 is empty, return l1
```

</> Code

C++ v Auto

```
1 class Solution {
2 public:
3     ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
4         if (l1 == NULL) return l2;
5         if (l2 == NULL) return l1;
6
7         if (l1->val > l2->val) swap(l1, l2);
8
9         ListNode* ans = l1;
10
11         while (l1 != NULL && l2 != NULL) {
12             ListNode* temp = NULL;
13             while (l1 != NULL && l1->val <= l2->val) {
14                 temp = l1;
```

Saved

Ln 18, Col 13

Testcase | Test Result

Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

Input

list1 =
[1,2,4]

list2 =
[1,3,4]