

Merge Two Sorted Lists

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

Description

Solution

Discuss (999+)

Submissions

Success

Details

Runtime: **8 ms**, faster than **74.64%** of C++ online submissions for Merge Two Sorted Lists.

Memory Usage: **14.9 MB**, less than **53.24%** of C++ online submissions for Merge Two Sorted Lists.

Next challenges:

[Merge k Sorted Lists](#)[Merge Sorted Array](#)[Sort List](#)

[Shortest Word Distance II](#)

[Add Two Polynomials Represented as Linked Lists](#)

[Longest Common Subsequence Between Sorted Arrays](#)

Show off your acceptance:

[f](#)[t](#)[in](#)

Time Submitted	Status	Runtime	Memory	Language
09/24/2021 16:13	Accepted	8 ms	14.9 MB	cpp

```
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* l1, ListNode* l2) {
14          ListNode *list = new ListNode();
15          ListNode *current = list;
16
17          while(l1 && l2){
18              if(l1->val < l2->val){
19                  ListNode *temp = l1;
20                  current->next = temp;
21                  current = temp;
22                  l1 = l1->next;
23              }else{
24                  ListNode *temp = l2;
25                  current->next = temp;
26                  current = temp;
27                  l2 = l2->next;
28              }
29          }
30
31          while(l1){
32              ListNode *temp = l1;
33              current->next = temp;
34              current = temp;
35              l1 = l1->next;
36          }
37
38          while(l2){
39              ListNode *temp = l2;
40              current->next = temp;
41              current = temp;
42              l2 = l2->next;
43          }
44
45          return list->next;
46      }
47  };
```

Reverse List

LeetCode

Explore

Problems

Interview

Contest

Discuss

Store

Description

Solution

Discuss (999+)

Submissions

Success

Details

Runtime: **4 ms**, faster than **40.32%** of C++ online submissions for Reverse Integer.

Memory Usage: **6.4 MB**, less than **7.15%** of C++ online submissions for Reverse Integer.

Next challenges:

[Reverse Bits](#)

Show off your acceptance:

[f](#)[t](#)[in](#)

Time Submitted	Status	Runtime	Memory	Language
09/24/2021 16:16	Accepted	4 ms	6.4 MB	cpp
10/02/2020 15:00	Accepted	32 ms	13.4 MB	python
10/02/2020 14:58	Time Limit Exceeded	N/A	N/A	python
10/02/2020 14:56	Wrong Answer	N/A	N/A	python
10/02/2020 14:54	Runtime Error	N/A	N/A	python

```
1  #include <string>
2  using namespace std;
3
4  class Solution {
5  public:
6      int reverse(int x) {
7          string str = "";
8          str += to_string(x);
9          string str2 = "";
10
11          if(x < 0){
12              str2 += "-";
13          }
14
15          for(int i = str.length()-1; i >= 0; i--){
16              if(str[i] == '-'){
17                  continue;
18              }else{
19                  str2 += str[i];
20              }
21          }
22
23          try{
24              return stoi(str2);
25          }catch(exception e){
26              return 0;
27          }
28      }
29  };
```

Valid parentheses

Success Details >

Runtime: 0 ms, faster than 100.00% of C++ online submissions for Valid Parentheses.

Memory Usage: 6.3 MB, less than 79.93% of C++ online submissions for Valid Parentheses.

Next challenges:

[Generate Parentheses](#) [Longest Valid Parentheses](#)
[Remove Invalid Parentheses](#) [Check If Word Is Valid After Substitutions](#)

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
09/24/2021 16:18	Accepted	0 ms	6.3 MB	cpp
10/03/2020 21:53	Accepted	40 ms	13.5 MB	python
10/03/2020 21:49	Wrong Answer	N/A	N/A	python

```
1 * #include <stack>
2 * using namespace std;
3 * class Solution {
4 * public:
5 *     bool isValid(string s) {
6 *         stack<char> stack;
7 *         if (s.size() <= 1) {
8 *             return false;
9 *         }
10 *         for (int i = 0; i < s.size(); i++) {
11 *             if (s[i] == '(' || s[i] == '[' || s[i] == '{') {
12 *                 stack.push(s[i]);
13 *                 continue;
14 *             }
15 *             else {
16 *                 if (stack.empty())
17 *                     return false;
18 *                 switch (s[i]) {
19 *                     case ')':
20 *                         if (stack.top() != '(')
21 *                             return false;
22 *                         break;
23 *                     case '}':
24 *                         if (stack.top() != '{')
25 *                             return false;
26 *                         break;
27 *                     case ']':
28 *                         if (stack.top() != '[')
29 *                             return false;
30 *                         break;
31 *                 }
32 *                 stack.pop();
33 *             }
34 *         }
35 *         if (!stack.empty())
36 *             return false;
37 *         return true;
38 *     }
39 * };
40 *
41 *
42 *
```

Palindrome Number

Success Details >

Runtime: 16 ms, faster than 40.33% of C++ online submissions for Palindrome Number.

Memory Usage: 6 MB, less than 30.87% of C++ online submissions for Palindrome Number.

Next challenges:

[Palindrome Linked List](#)

Show off your acceptance: [f](#) [t](#) [in](#)

Time Submitted	Status	Runtime	Memory	Language
09/24/2021 16:17	Accepted	16 ms	6 MB	cpp
10/15/2020 17:25	Accepted	32 ms	13.4 MB	python

```
1 * #include <string>
2 * using namespace std;
3 *
4 * class Solution {
5 * public:
6 *     bool isPalindrome(int x) {
7 *         if (x < 0)
8 *             return false;
9 *
10 *         string str = "";
11 *         str += to_string(x);
12 *         string str2 = "";
13 *
14 *         for (int i = str.length() - 1; i >= 0; i--) {
15 *             if (str[i] == '-')
16 *                 continue;
17 *             else {
18 *                 str2 += str[i];
19 *             }
20 *         }
21 *         return str.compare(str2) == 0;
22 *     }
23 * };
24 *
25 *
26 *
27 *
```