

Description

 Editorial 

Solutions (8.7K)

Submissi...

i Java 

Auto

 Accepted

Runtime

Details

Memory

1 ms

43.19 mb

Beats 100.00% of users with Java

Beats 89.18%

Next question


24. Swap Nodes in Pairs


More challenges

21. Merge Two Sorted Lists

264. Ugly Number I

2411. Smallest Subarrays With Maximum Bitwise OR

All statuses 

All languages 

Tim...

Accepted

Java

in a

Compile Error

C++

12

Accepted

Java

an

```
1 class Solution {
2     public ListNode mergeKLists(ListNode[] lists) {
3         if (lists == null || lists.length == 0)
4             return null;
5     }
6     return mergeLists(lists, 0, lists.length - 1);
7 }
8
9 private ListNode mergeLists(ListNode[] lists, int left, int right) {
10     if (left == right) {
11         return lists[left];
12     }
13     int mid = left + (right - left) / 2;
14     ListNode leftList = mergeLists(lists, left, mid);
15     ListNode rightList = mergeLists(lists, mid + 1, right);
16     return mergeTwoLists(leftList, rightList);
17 }
18
19 private ListNode mergeTwoLists(ListNode l1, ListNode l2) {
20     ListNode dummy = new ListNode(0);
21     ListNode current = dummy;
22
23     while (l1 != null && l2 != null) {
24         if (l1.val < l2.val) {
25             current.next = l1;
26             l1 = l1.next;
27         } else {
28             current.next = l2;
29             l2 = l2.next;
30         }
31         current = current.next;
32     }
33
34     if (l1 != null) {
35         current.next = l1;
36     }
37
38     if (l2 != null) {
39         current.next = l2;
40     }
41
42     return dummy.next;
43 }
44 }
```