Name: Anshul

UID: 22BCS16477

Section/Group: 609(B)

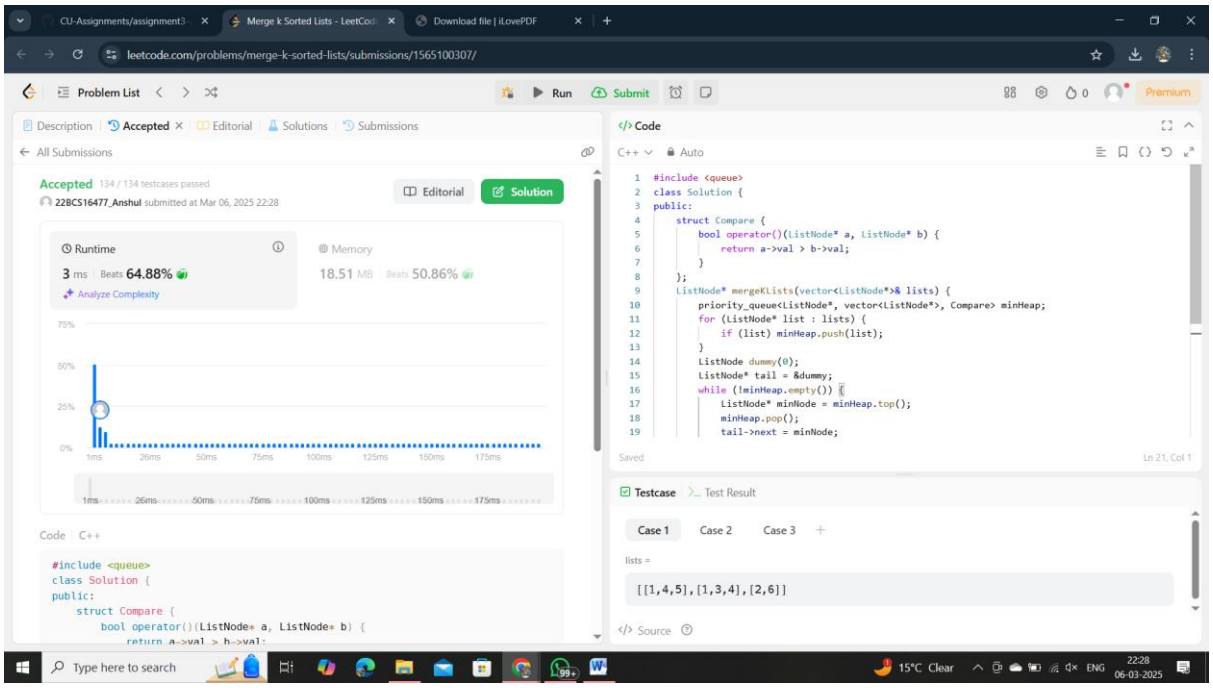
Merge k Sorted Lists

Code:

```
#include <queue>
class Solution {
public:
    struct Compare {
        bool operator()(ListNode* a, ListNode* b) {
            return a->val > b->val;
        }
    };
    ListNode* mergeKLists(vector<ListNode*>& lists) {
        priority_queue<ListNode*, vector<ListNode*>, Compare> minHeap;
        for (ListNode* list : lists) {
            if (list) minHeap.push(list);
        }
        ListNode dummy(0);
        ListNode* tail = &dummy;
        while (!minHeap.empty()) {
            ListNode* minNode = minHeap.top();
            minHeap.pop();
            tail->next = minNode;
            tail = minNode;

            if (minNode->next) {
                minHeap.push(minNode->next);
            }
        }
        return dummy.next;
    }
};
```

Output:


Merge k Sorted Lists - LeetCod
Download file | iLovePDF
leetcode.com/problems/merge-k-sorted-lists/submissions/1565100307/
Problem List
Run
Submit
Premium
Description
Accepted
Editorial
Solutions
Submissions
All Submissions
Accepted 134 / 134 testcases passed
22BCS16477_Anshul submitted at Mar 06, 2025 22:28
Editorial
Solution
Runtime
3 ms | Beats 64.88%
Analyze Complexity
Memory
18.51 MB Beats 50.86%
Code
C++
#include <queue>
class Solution {
public:
struct Compare {
bool operator()(ListNode* a, ListNode* b) {
return a->val > b->val;
}
};
ListNode* mergeKLists(vector<ListNode*>& lists) {
priority_queue<ListNode*, vector<ListNode*>, Compare> minHeap;
for (ListNode* list : lists) {
if (list) minHeap.push(list);
}
ListNode dummy(0);
ListNode* tail = &dummy;
while (!minHeap.empty()) {
ListNode* minNode = minHeap.top();
minHeap.pop();
tail->next = minNode;
Saved
Ln 21, Col 1
Testcase
Test Result
Case 1
Case 2
Case 3
lists =
[[1,4,5],[1,3,4],[2,6]]
Source
Type here to search
15°C Clear
ENG
22:28
06-03-2025