Problem 23: Merge k Sorted Lists

Problem Information

Difficulty: Hard
Acceptance Rate: 58.16%
Paid Only: No
Tags: Linked List, Divide and Conquer, Heap (Priority Queue), Merge Sort

Problem Description

You are given an array of `k` linked-lists `lists`, each linked-list is sorted in ascending order.

Merge all the linked-lists into one sorted linked-list and return it.

Example 1:

Input: lists = [[1,4,5],[1,3,4],[2,6]] **Output:** [1,1,2,3,4,4,5,6] **Explanation:** The linked-lists are: [1->4->5, 1->3->4, 2->6] merging them into one sorted linked list: 1->1->2->3->4->4->5->6

Example 2:

Input: lists = [] **Output:** []

Example 3:

Input: lists = [[]] **Output:** []

Constraints:

* `k == lists.length` * `0 <= k <= 104` * `0 <= lists[i].length <= 500` * `-104 <= lists[i][j] <= 104` * `lists[i]` is sorted in **ascending order**. * The sum of `lists[i].length` will not exceed `104`.

Code Snippets

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) {}
* };
*/
class Solution {
public:
ListNode* mergeKLists(vector<ListNode*>& lists) {

}
};
```

Java:

```
/**
* Definition for singly-linked list.
* public class ListNode {
* int val;
* ListNode next;
* ListNode() {}
* ListNode(int val) { this.val = val; }
* ListNode(int val, ListNode next) { this.val = val; this.next = next; }
* }
*/
class Solution {
public ListNode mergeKLists(ListNode[] lists) {

}
}
```

Python3:

```
# Definition for singly-linked list.
# class ListNode:
# def __init__(self, val=0, next=None):
# self.val = val
```

```
# self.next = next
class Solution:
def mergeKLists(self, lists: List[Optional[ListNode]]) -> Optional[ListNode]:
```