

合并 K 个升序链表

给你一个链表数组，每个链表都已经按升序排列。

请你将所有链表合并到一个升序链表中，返回合并后的链表。

示例 1：

输入：lists = [[1,4,5],[1,3,4],[2,6]]

输出：[1,1,2,3,4,4,5,6]

解释：链表数组如下：

```
[  
  
    1->4->5,  
  
    1->3->4,  
  
    2->6  
]
```

将它们合并到一个有序链表中得到。

1->1->2->3->4->4->5->6

示例 2：

输入：lists = []

输出：[]

示例 3：

输入：lists = [[]]

输出：[]

```
/**  
 * 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:
    //以合并两个有序链表为基础
    ListNode* mergeTwoLists(ListNode* l1, ListNode* l2) {
        ListNode* preHead = new ListNode(-1);

        ListNode* prev = preHead;
        while (l1 != nullptr && l2 != nullptr) {
            if (l1->val < l2->val) {
                prev->next = l1;
                l1 = l1->next;
            } else {
                prev->next = l2;
                l2 = l2->next;
            }
            prev = prev->next;
        }

        // 合并后 l1 和 l2 最多只有一个还未被合并完，我们直接将链表末尾指向未合并
        // 完的链表即可
        prev->next = l1 == nullptr ? l2 : l1;

        return preHead->next;
    }

    ListNode* mergeKLists(vector<ListNode*> lists) {
        ListNode* res=nullptr;
        for(int i=0;i<lists.size();i++)
        {
            res=mergeTwoLists(res,lists[i]);
        }
        return res;
    }
};

```