23. Merge k Sorted Lists

Hard

1942124FavoriteShare

Merge *k* sorted linked lists and return it as one sorted list. Analyze and describe its complexity.

Example:

Input:  
[  
 1->4->5,  
 1->3->4,  
 2->6  
]  
Output: 1->1->2->3->4->4->5->6

/\*\*

\* Definition for singly-linked list.

\* struct ListNode {

\* int val;

\* ListNode \*next;

\* ListNode(int x) : val(x), next(NULL) {}

\* };

\*/

class Solution {

public:

struct MyComparator{

bool operator()(ListNode\* x, ListNode\* y){

return x->val>y->val;

}

};

ListNode\* mergeKLists(vector<ListNode\*>& lists) {

if(lists.size()==0) return NULL;

priority\_queue<ListNode\*, vector<ListNode\*>, MyComparator> pq;

ListNode\* dummy=(ListNode\*)malloc(sizeof(struct ListNode));

ListNode\* tail=dummy;

for(int i=0;i<lists.size();i++){

if(lists[i]!=NULL) pq.push(lists[i]);

}

while(pq.empty()==false){

ListNode\* curr=pq.top();

pq.pop();

if(curr->next!=NULL) pq.push(curr->next);

tail->next=curr;

tail=tail->next;

}

return dummy->next;

}

};

Success

[Details](https://leetcode.com/submissions/detail/205538931/)

Runtime: 16 ms, faster than 98.56% of C++ online submissions for Merge k Sorted Lists.

Memory Usage: 1.9 MB, less than 63.07% of C++ online submissions forMerge k Sorted Lists.