Merge k Sorted Lists

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
```

Solution in C++

```
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution {
public:
    int GetMinIndex(vector<ListNode *>& lists) {
        int iMin=-1;
        for(int i=0; i<lists.size(); i++) if (lists[i]!=NULL) {
            if (iMin==-1) iMin=i;
            else if (lists[iMin]->val>lists[i]->val) iMin=i;
        }
        return iMin;
    }

    ListNode* mergeKLists(vector<ListNode*>& lists) {
        ListNode *h=NULL, *p;

        int iMin=GetMinIndex(lists);
        if (iMin==-1) return h;

        p=h=new ListNode(lists[iMin]->val);
        lists[iMin]=lists[iMin]->next;

        while((iMin=GetMinIndex(lists))!=-1) {
            p->next=new ListNode(lists[iMin]->val);
            p=p->next;
            lists[iMin]=lists[iMin]->next;
        }

        return h;
    }
};
```