```c
/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     struct ListNode *next;
 * };
 */

struct ListNode* mergeKLists(struct ListNode** lists, int listsSize){
    struct ListNode* tmp;
    struct ListNode* node_pool;
    int* map;
    int max;
    int min;
    int index;
    int count;
    int count_index;
    int map_length;

    max = INT_MIN;
    min = INT_MAX;
    count = 0;
    node_pool = NULL;

    for(index = 0; index < listsSize; index++)
    {
        tmp = lists[index];

        while(tmp != NULL)
        {
            if(tmp->val < min)
            {
                min = tmp->val;
            }

            if(tmp->val > max)
            {
                max = tmp->val;
            }

            tmp = tmp->next;
            count++;
        }
    }

    if(count)
    {
        map_length = max - min + 1;
        map = (int*)calloc(sizeof(int), map_length);
        node_pool = (struct ListNode*)malloc(sizeof(struct ListNode) * count);

        for(index = 0; index < listsSize; index++)
        {
            tmp = lists[index];

            while(tmp != NULL)
            {
                map[tmp->val - min]++;
                tmp = tmp->next;
            }
        }

        count_index = 0;

        for(index = 0; index < map_length; index++)
        {
            while(map[index]>0)
            {
                node_pool[count_index].val = index + min;

                if(count_index < (count-1))
                {
                    node_pool[count_index].next = &node_pool[count_index+1];
                }else
                {
                    node_pool[count_index].next = NULL;
                }


                count_index++;
                map[index]--;
            }
        }

    }

    return node_pool;

}
```