一、题目说明

这个题目是21. Merge Two Sorted Lists，归并2个已排序的列表。难度是Easy！

二、我的解答

既然是简单的题目，应该一次搞定。确实1次就搞定了，但是性能太差：

```
Runtime: 20 ms, faster than 8.74% of C++ online submissions for Merge Two Sorted
Lists.
Memory Usage: 9.4 MB, less than 5.74% of C++ online submissions for Merge Two
Sorted Lists.
```

代码如下：

```
class Solution{
    public:
        ListNode* mergeTwoLists(ListNode* l1, ListNode* l2){
            if(l1 ==NULL && l2==NULL) return NULL;
            if(l1 !=NULL && l2==NULL) return l1;
            if(l1 ==NULL && l2!=NULL) return l2;

            ListNode dummy(-1);
            ListNode* p = &dummy;
            while(l1 !=NULL && l2 !=NULL){
                if(l1->val <= l2->val){
                     p->next = l1;
                     p = p->next;
                     l1 = l1->next;
                }else{
                    p->next = l2;
                    p = p->next;
                    l2 = l2->next;
                }
            }

            if(l1 !=NULL){
                p->next = l1;
            }

            if(l2 !=NULL){
                p->next = l2;
            }

            return dummy.next;
        }
};
```

三、优化措施

优化后，8s，代码如下：

```
#include<iostream>
```

```
using namespace std;
struct ListNode{
    int val;
    ListNode*next;
    ListNode(int x):val(x),next(NULL){
    }
};

/**
 * Definition for singly-linked list.
 * struct ListNode {
 *     int val;
 *     ListNode *next;
 *     ListNode(int x) : val(x), next(NULL) {}
 * };
 */
class Solution{
    public:
        ListNode* mergeTwoLists(ListNode* l1, ListNode* l2){
            if(NULL == l1) return l2;
            if(NULL == l2) return l1;

            ListNode dummy(-1);
            ListNode* p = &dummy;
            while(l1  && l2 ){
                if(l1->val <= l2->val){
                     p->next = l1;
                     l1 = l1->next;
                }else{
                    p->next = l2;
                    l2 = l2->next;
                }
                p = p->next;
            }

            p->next = l1 ? l1 : l2;

            return dummy.next;
        }
};

int main(){
    Solution s;
    ListNode* l1,*l2;
    ListNode* tmp;

    //init l1
    tmp = new ListNode(4);
    l1 = tmp;

    tmp = new ListNode(2);
    tmp->next = l1;
    l1 = tmp;

    tmp = new ListNode(1);
    tmp->next = l1;
    l1 = tmp;
    //init l2
    tmp = new ListNode(4);
    l2 = tmp;

    tmp = new ListNode(3);
    tmp->next = l2;
    l2 = tmp;

    tmp = new ListNode(1);
    tmp->next = l2;
    l2 = tmp;


    ListNode* l3 = s.mergeTwoLists(l1,l2);
    while(l3!=NULL){
        cout<<l3->val<<" ";
        l3 = l3->next;
    }
    return 0;
}
```