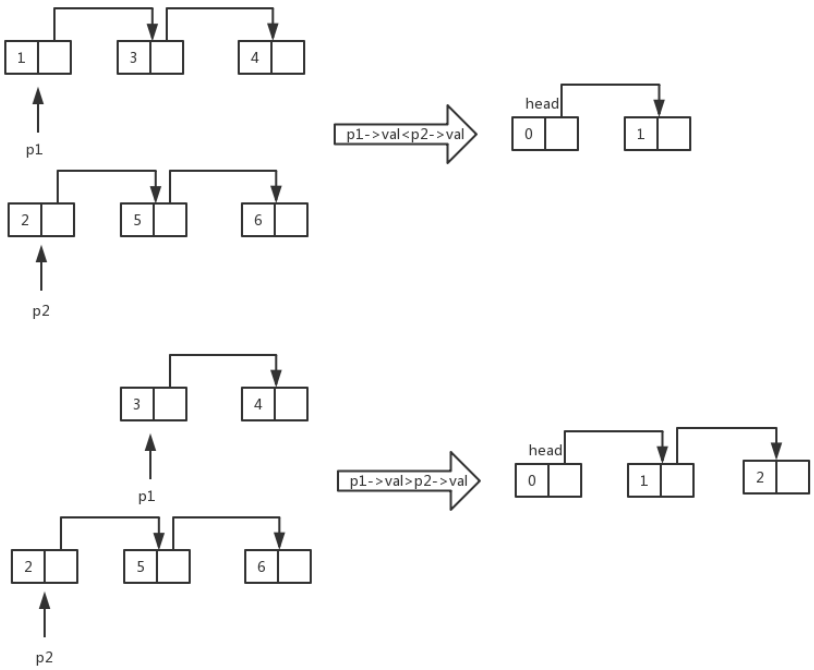
LeetCode - Merge Two Sorted Lists

Merge two sorted linked lists and return it as a new list. The new list should be made by splicing together the nodes of the first two lists.

将两个有序的链表合并为一个有序的链表。

由于给出的两个链表l_1,l_2是有序的（从小到大排序），我们只需要依次拿出两个链表最前端的数比较，取较小值为新链表l_3的下一个结点并在原来的链表中删除该结点。重复这个行为，直到l_1,l_2其中一个链表为空，此时将不为空的另一个链表加入l_3中。

我们用图像来简单的表示这个问题：（上面的链表为l_1,下面的链表为l_2）



为新的链表设置一个头结点head，令两个指针分别指向当前两个链表中需要对比的值，令较小值为新链表的下一个结点值。当然新链表也需要一个指针不断指向下一个地址（不过我图上忘记画了，手动笑脸 :-D）。头结点的值是不重要的，可以不赋值，在一定要初始化的情况下往往赋值为-1或者0。

下面给出代码：

```cpp
/** C++
 * 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)
    {
        ListNode head(0);
        ListNode *cur = &head;
        while(l1 && l2)
        {
            if(l1->val > l2->val)
            {
            cur->next = l2;
            l2 = l2->next;
            }
            else
            {
            cur->next = l1;
            l1 = l1->next;
            }
            cur = cur->next;
        }
        if(l1==NULL)
        {
           cur->next = l2;
        }
        else
        {
            cur->next = l1;
        }
        return head.next;
    }

};


```