Merge Two sorted list

BRUTE FORCE :

Merge two list by creating a new list comparing list1 and list2 value.

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if(list1==NULL)
            return list2;
        if(list2==NULL)
            return list1;
        ListNode* head=NULL;
        if(head==NULL)
        {
            if(list1->val<=list2->val)
            {
                ListNode* newNode=new ListNode(list1->val);
                head=newNode;
                list1=list1->next;
            }
            else
            {
                ListNode* newNode=new ListNode(list2->val);
                head=newNode;
                list2=list2->next;
            }

        }
        ListNode* temp=head;
        while(list1!=NULL && list2!=NULL)
        {
            ListNode* newNode=NULL;
            if(list1->val<list2->val)
            {

                newNode=new ListNode(list1->val);
                temp->next=newNode;
                list1=list1->next;
            }
            else
            {
                newNode=new ListNode(list2->val);
                temp->next=newNode;
                list2=list2->next;
            }
            temp=temp->next;
```

```
        }
        while(list1!=NULL)
        {
            temp->next=list1;
            list1=list1->next;
            temp=temp->next;
        }
        while(list2!=NULL)
        {
            temp->next=list2;
            list2=list2->next;
            temp=temp->next;
        }
        return head;

    }
};
```

- Time Complexity : O(N1+N2)
- Space Complexity : O(N1+N2)

Optimal Approach

Reducing space using in place merging. We will always point list1 to the shorter value if we encounter a greater element in list 1 we always keep a track of previous element and point the previous element next to list2 and then swap list1 and list 2 so that list1 now points to a smaller element. Two while loops are used but each elem is traversed only once.

```
class Solution {
public:
    ListNode* mergeTwoLists(ListNode* list1, ListNode* list2) {
        if(list1==NULL)
            return list2;
        if(list2==NULL)
            return list1;
        if(list1->val > list2->val)
            swap(list1,list2);
        ListNode* head=list1;
        ListNode* prev=new ListNode();
        while(list1!=NULL&& list2!=NULL)
        {
            while(list1!=NULL && list1->val <= list2->val)
            {
                prev=list1;
```

```
                list1=list1->next;
            }
            prev->next=list2;
            swap(list1,list2);

        }
        return head;

    }
};
```

- Time Complexity : O(N1+N2)
- Space Complexity : O(1)