

学号	2022212080	算法思路 (30%)	编码实现与 算法分析 (50%)	实验报告 (20%)	得分
姓名	刘纪彤				
评语					

《算法设计与分析》实验报告

实验 1 递归算法实验

一、实验目的

1. 加深对递归算法的理解，包括递归的基本原理和递归调用的机制。
2. 学习如何设计和实现递归函数来解决具体问题。
3. 掌握递归算法的终止条件和递归步骤。
4. 分析递归算法的时间复杂度和空间复杂度。
5. 提高编程能力和问题解决能力。

二、实验内容(题目)

1. 用递归法实现插入排序。
2. 用递归法实现单链表的逆转。

三、算法设计思路

插入排序：利用递归的知识，我们通过对 1-size 的所有元素插入到前方已经排好序的数组中，最后当他大于 size 的时候跳出递归。

单链表逆转：当他链表为空或者下一个为空的时候我们将这个结点作为头节点返回，同时实现逆转的就是他的下一个的下一个指向他，循环往复

四、各功能模块设计

```
#include <bits/stdc++.h>
using namespace std;
void insert_sort(vector<int>&nums,int index)
{
    //因为要以 index 为基准，所以 index 不能小于 1，不能大于 nums.size()
    if(index<1) return;
    if(index>=(int)nums.size()) return; //递归出口出口
    //设置 temp 为当前 index 的值
    int temp=nums[index];
    int j=index-1;
    while(j>=0&&temp<nums[j])
```

```

{
    //循环将大于 temp 的值后移
    nums[j+1]=nums[j];
    j--;
}
//将 temp 插入到合适的位置
nums[j+1]=temp;
insert_sort(nums,index+1);
}
int main()
{
    vector<int> nums={3,2,1,4,5,6,7,8,3,21,9,2};
    insert_sort(nums,1);
    for(auto i:nums)
    {
        cout<<i<<" ";
    }
}

```

对于插入排序的时间复杂度因为他要对整个数组进行遍历，还需要每个数据进行比较，故其最坏的时间复杂度可以是

$$\sum_{i=1}^n n = \frac{n*(n-1)}{2}$$

可以表示为 $O(n^2)$ ，最好情况就是已经有序复杂度就是 $O(n)$ ，平均下来也要达到 $O(n^2)$

```

#include<bits/stdc++.h>
using namespace std;
class ListNode
{
public:
    int val;
    ListNode* next;
    ListNode(int val):val(val),next(nullptr){}
    ListNode(int val,ListNode* next):val(val),next(next){}
};
ListNode* reserveLinkedList(ListNode* head)
{
    if(head==nullptr || head->next==nullptr) return head; // 如果链表为空或者只有一个节点，直接返回
    ListNode* newHead = reserveLinkedList(head->next); // 翻转以 head->next 为头的链表
    head->next->next = head; // 将 head 节点连接到翻转后的链表的尾部
    head->next = nullptr; // 将 head 节点的 next 指针置空
    return newHead; // 返回新的头节点
}

```

```

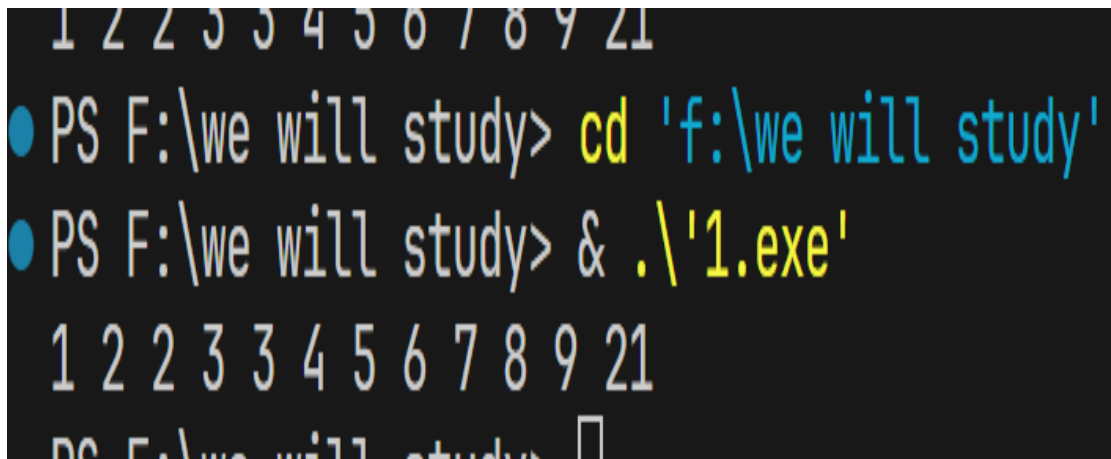
}
int main()
{
    ListNode* head=new ListNode(1,new ListNode(2,new ListNode(3,new ListNode(4,new
ode(4,new ListNode(5))));
    ListNode* newHead=reserveLinkedList(head);
    while(newHead!=nullptr)
    {
        cout<<newHead->val<<" ";
        newHead=newHead->next;
    }
}

```

链表翻转只需要遍历整个链表即可，故时间复杂度为 $O(n)$

五、运行结果与分析

(运行结果截图，并说明其正确)

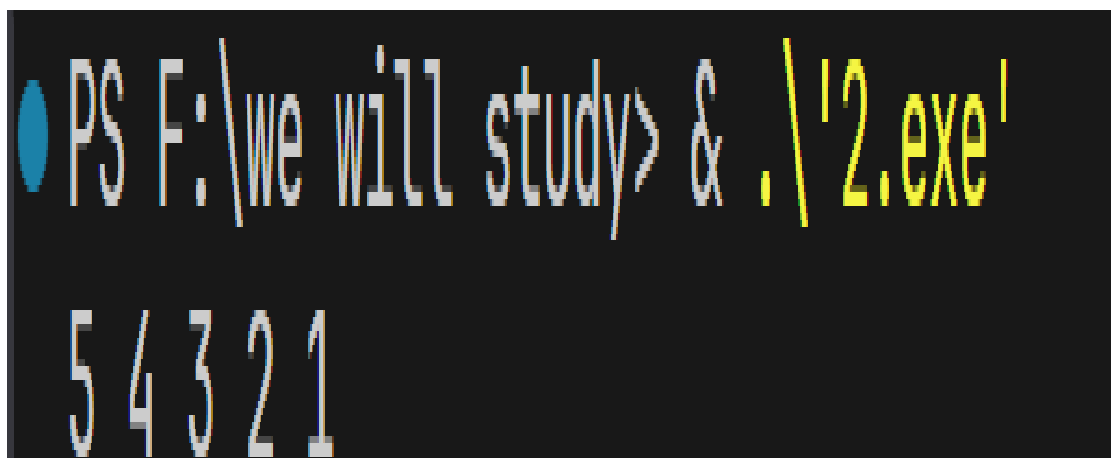


```

1 2 2 3 3 4 5 6 7 8 9 21
PS F:\we will study> cd 'f:\we will study'
PS F:\we will study> & .\'1.exe'
1 2 2 3 3 4 5 6 7 8 9 21
PS F:\we will study>

```

1-1



```

PS F:\we will study> & .\'2.exe'
5 4 3 2 1

```

1-2

如图所示 1-1 按升序排列好，1-2 按照倒转顺序也正确，故本运行结果正确

六、实验总结

通过本次实验我已经了解并掌握了递归的基本逻辑,能够使用递归的思路解决实际问题,受益匪浅。