2021/3/19

1. cd dir
2. cd..
3. Tab补全文件
4. Python基础
5. 函数

2021/3/29

1. windows+d 显示桌面
2. 面向对象编程：封装，继承，多态。
3. 简单的说，类是对象的蓝图和模板，而对象是类的实例。这个解释虽然有点像用概念在解释概念，但是从这句话我们至少可以看出，类是抽象的概念，而对象是具体的东西。在面向对象编程的世界中，一切皆为对象，对象都有属性和行为，每个对象都是独一无二的，而且对象一定属于某个类（型）。当我们把一大堆拥有共同特征的对象的静态特征（属性）和动态特征（行为）都抽取出来后，就可以定义出一个叫做“类”的东西。
4. # \_\_init\_\_是一个特殊方法用于在创建对象时进行初始化操作  
   # 通过这个方法我们可以为学生对象绑定name和age两个属性

2021/4/1

[21. 合并两个有序链表](https://leetcode-cn.com/problems/merge-two-sorted-lists/)

# Definition for singly-linked list.

# class ListNode:

#     def \_\_init\_\_(self, val=0, next=None):

#         self.val = val

#         self.next = next

class Solution:

    def mergeTwoLists(self, l1: ListNode, l2: ListNode) -> ListNode:

        if not l1:

            return l2

        elif not l2:

            return l1

        elif l1.val <= l2.val:

            l1.next = self.mergeTwoLists(l1.next,l2)

            return l1

        else:

            l2.next = self.mergeTwoLists(l1,l2.next)

            return l2

[83. 删除排序链表中的重复元素](https://leetcode-cn.com/problems/remove-duplicates-from-sorted-list/)

# Definition for singly-linked list.

# class ListNode:

#     def \_\_init\_\_(self, val=0, next=None):

#         self.val = val

#         self.next = next

class Solution:

    def deleteDuplicates(self, head: ListNode) -> ListNode:

        if not head:

            return head

        cur = head

        while cur.next:

            if cur.val == cur.next.val:

                cur.next = cur.next.next

            else:

                cur = cur.next

        return head

链表去重、删除等算是最基础的算法，实在没什么好分析的，那不如将链表去重比喻成奴隶主的故事。

从前有个奴隶主，他有一根铁链(链表)，拴着自己管理的奴隶们。

现在他要花钱找了一个打手cur，帮自己管教一下身后的一堆奴隶。

但如果奴隶主一回头，发现铁链上一个奴隶都没有，那算了，我回家自己玩去吧。

即：if not head: return head

如果奴隶发现他的铁链上拴着一堆奴隶，就可以把这个cur的打手请来了。

接着奴隶主他把钥匙(指针)给cur说：你去收拾我身后的奴隶们吧。

发现不听话的，拿钥匙解开他的铁链暴打一顿，关到黑屋，

然后把拴着他的链子接到下一个人身上（指针指向下一个人）！

cur拿着顺着铁链从一个个奴隶主身边走过，发现欠收拾(不满足题意)的，

暴打一顿然后把他从铁链上卸下来，接着找下一个。

一直这样找到最后，发现该收拾的都收拾了，此时发现打的人太多钥匙都扭断了(指针结束)。

奴隶主需要检验下cur管教的效果如何，顺着铁链走一遍，没问题，AC了...

恭喜奴隶主，每日一题通过...

作者：qingfengpython

链接：https://leetcode-cn.com/problems/remove-duplicates-from-sorted-list/solution/83-shan-chu-pai-xu-lian-biao-zhong-de-zh-auj4/

来源：力扣（LeetCode）

著作权归作者所有。商业转载请联系作者获得授权，非商业转载请注明出处。