

Algorithm Homework 3

姓名：阚建明

学号：1901213139

作业要求：

Leetcode: 35,58,83

题解：

• Leetcode problem 35 ([搜索插入位置](#))

◦ 算法思路：

- 尝试使用二分法找到target的索引，找到则返回
- 没有找到则判断target与左右索引的大小关系，返回应该插入的位置

◦ Leetcode提交结果：

执行结果： 通过 [显示详情](#)

执行用时： **28 ms** ，在所有 Python3 提交中击败了 **99.35%** 的用户

内存消耗： **14 MB** ，在所有 Python3 提交中击败了 **5.02%** 的用户

炫耀一下：



◦ 代码：

```
1 class Solution:
2     def searchInsert(self, nums: List[int], target:
3         int) -> int:
4         left, right = 0, len(nums) - 1
5         while left < right - 1:
6             pos = (left + right) // 2
7             if nums[pos] > target:
8                 right = pos
9             elif nums[pos] < target:
10                left = pos
11            else:
```

```

11         return pos
12         if nums[left] == target or target <
nums[left]:
13             return left
14         elif nums[right] == target or target <
nums[right]:
15             return right
16         else:
17             return right + 1

```

• Leetcode problem 58 (最后一个单词的长度)

◦ 算法思路：

- 思路1：使用 `split` 将字符串用空格分割成数组，然后反序遍历，取第一个非空字符串的长度，无则返回0；
- 思路2：使用 `strip` 截掉字符串前后的空格，然后反序遍历统计最后一个单次的长度，无则返回0。

◦ Leetcode提交结果：

执行结果： 通过 [显示详情 >](#)

执行用时： **28 ms** ，在所有 Python3 提交中击败了 **90.55%** 的用户

内存消耗： **13.5 MB** ，在所有 Python3 提交中击败了 **5.81%** 的用户

炫耀一下：



◦ 代码：

```

1  # 思路1代码
2  class Solution:
3      def lengthOfLastWord(self, s: str) -> int:
4          strs = s.split(' ')
5          for i in range(0, len(strs))[::-1]:
6              if len(strs[i]) > 0: return len(strs[i])
7          return 0
8
9  # 思路2代码（性能更优）
10 class Solution:
11     def lengthOfLastWord(self, s: str) -> int:
12         s = s.strip()
13         count = 0
14         for i in range(0, len(s))[::-1]:

```

```
15         if s[i] == ' ':
16             return count
17         else:
18             count += 1
19     return count
```

• Leetcode problem 83 ([删除排序链表中的重复元素](#))

◦ 算法思路：

- 用一个set记录当前遍历过的值
- 遍历到每一个节点，用set检查当前值是否存在于set，如果存在则删除该节点

◦ Leetcode提交结果：

执行结果： **通过** [显示详情](#)

执行用时： **36 ms** ，在所有 Python3 提交中击败了 **96.78%** 的用户

内存消耗： **13.7 MB** ，在所有 Python3 提交中击败了 **5.23%** 的用户

炫耀一下：



◦ 代码：

```
1 class Solution:
2     def deleteDuplicates(self, head: ListNode) ->
   ListNode:
3         if head == None:
4             return head
5         s, pre, cur = set(), head, head.next
6         s.add(pre.val)
7         while cur is not None:
8             if cur.val in s:
9                 pre.next, cur = cur.next, cur.next
10                continue
11            else:
12                s.add(cur.val)
13                pre, cur = cur, cur.next
14        return head
```