

实验三排序算法

2023 年 3 月 16 日

1 实验三排序算法

1.1 设计思路

给定 n 个学生，以及其信息，分析其最多奖学金的姓名，以及奖金数额，和总奖金。

- 设定一个学生主体，它的属性包含期末平均成绩，论文数，班级评议成绩，是否为西部省份学生，是否为学生干部。
- 编写 5 个判断函数，对于每个学生，分别执行判断，并将奖学金总加进 sum 里面
- 然后用 sort 函数进行排序每个学生的奖学金，一定注意要稳定排序，才能保证特殊情况下最高奖学金中是输入文件中出现最早的学生的姓名，在此我仅仅采用 python 官方 sort 函数，归并也可既快又稳定。

1.2 伪代码与思路

先设定一个结构体，名字为 student，属性为各个条件，然后结构体函数就是 5 个判断奖学金。然后调用 sort 配合 lambda 将奖学金拉出来比较。

以下为伪代码：

```
[ ]: CLASS student
    FUNCTION INIT(self, input_string)
        info = input_string.split()
        self.name = info[0]
        self.final_grade = int(info[1])
        self.evaluate_grade = int(info[2])
        self.is_leader = True IF info[3] == "Y" ELSE False
        self.is_westStu = True IF info[4] == "Y" ELSE False
        self.paper = int(info[5])
        self.scholarship = 0
    END FUNCTION
```

```

FUNCTION cal_scholarship(self)
    IF self.final_grade > 80 AND self.paper >= 1
        self.scholarship += 8000
    END IF
    IF self.final_grade > 85 AND self.evaluate_grade > 80
        self.scholarship += 4000
    END IF
    IF self.final_grade > 90
        self.scholarship += 2000
    END IF
    IF self.final_grade > 85 AND self.is_westStu
        self.scholarship += 1000
    END IF
    IF self.evaluate_grade > 80 AND self.is_leader
        self.scholarship += 850
    END IF
END FUNCTION
END CLASS

n = int(input())
arr = []
total = 0

FOR i = 0 TO n-1
    stu = student(input())
    stu.cal_scholarship()
    total += stu.scholarship
    arr.append(stu)
END FOR

arr.sort(key=lambda x: x.scholarship,reverse=True)

PRINT arr[0].name
PRINT arr[0].scholarship
PRINT total

```

1.3 python 代码实现

```
[ ]: class student:
    def __init__(self, input_string) -> None:
        info = input_string.split()
        self.name = info[0]
        self.final_grade = int(info[1])
        self.evaluate_grade = int(info[2])
        self.is_leader = True if info[3] == "Y" else False
        self.is_westStu = True if info[4] == "Y" else False
        self.paper = int(info[5])
        self.scholarship = 0
    def cal_scholarship(self):
        if self.final_grade > 80 and self.paper >= 1:
            self.scholarship += 8000
        if self.final_grade > 85 and self.evaluate_grade > 80:
            self.scholarship += 4000
        if self.final_grade > 90:
            self.scholarship += 2000
        if self.final_grade > 85 and self.is_westStu:
            self.scholarship += 1000
        if self.evaluate_grade > 80 and self.is_leader:
            self.scholarship += 850

n = int(input())
arr = []
total = 0
for i in range(n):
    stu = student(input())
    stu.cal_scholarship()
    total += stu.scholarship
    arr.append(stu)

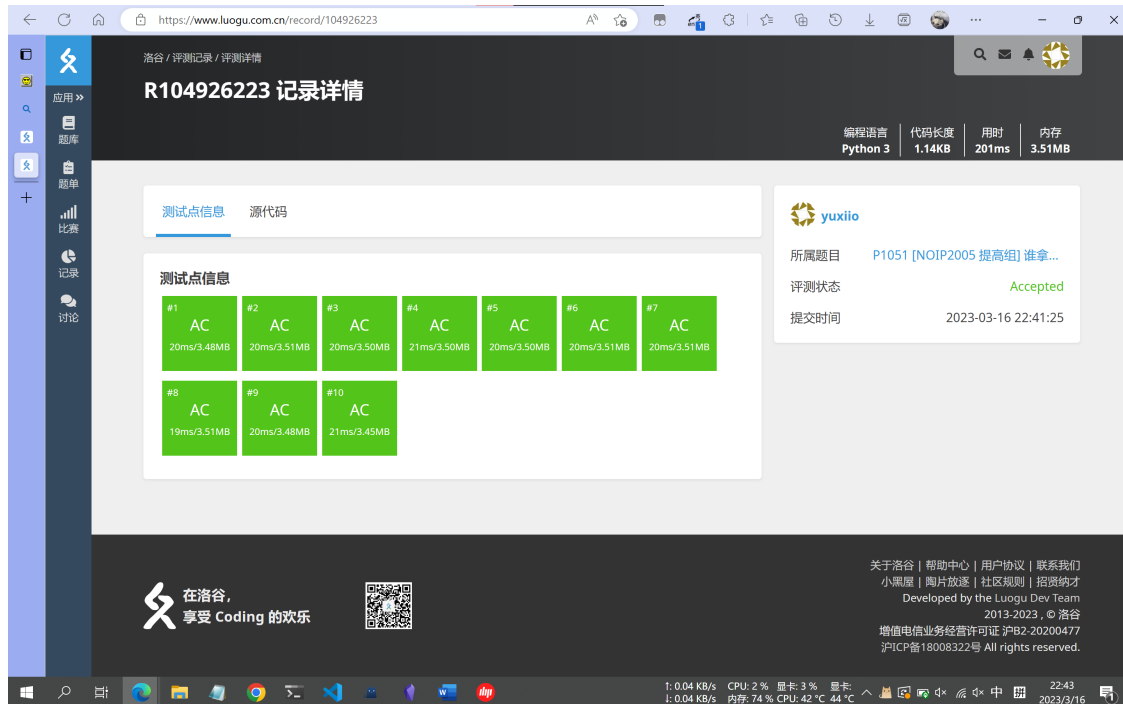
arr.sort(key=lambda x: x.scholarship, reverse=True)

print(arr[0].name)
print(arr[0].scholarship)
```

```
print(total)
```

1.4 代码运行情况

已通过洛谷测试



洛谷 / 评测记录 / 评测详情

R104926223 记录详情

编程语言: Python 3 | 代码长度: 1.14KB | 用时: 201ms | 内存: 3.51MB

所属题目: P1051 [NOIP2005 提高组] 谁拿...
评测状态: Accepted
提交时间: 2023-03-16 22:41:25

测试点信息

测试点	状态	用时	内存
#1	AC	20ms/3.48MB	
#2	AC	20ms/3.51MB	
#3	AC	20ms/3.50MB	
#4	AC	21ms/3.50MB	
#5	AC	20ms/3.50MB	
#6	AC	20ms/3.51MB	
#7	AC	20ms/3.51MB	
#8	AC	19ms/3.51MB	
#9	AC	20ms/3.48MB	
#10	AC	21ms/3.45MB	

在洛谷，
享受 Coding 的快乐

关于洛谷 | 帮助中心 | 用户协议 | 联系我们
小黑屋 | 图片放逐 | 社区规则 | 招贤纳士
Developed by the Luogu Dev Team
2013-2023, © 洛谷
增值电信业务经营许可证 沪B2-20200477
沪ICP备18008322号 All rights reserved.

1: 0.04 KB/s CPU: 2% 显卡: 3% 温度: 22.43
1: 0.04 KB/s 内存: 74% CPU: 42 °C 44 °C 2023/3/16

1.5 总结

本次实验是一个简单的储存条件，判断条件，进行排序的基础题。

之前遇到这道题时用 C 处理，现在改成 python。发现 python 的类与 C 的结构体的不同，甚至和 Java 的类也不同。因为不能储存变量，只能在构造方法上进行设置。算是领略到了不同语言的风格和语法糖。其次，再一次使用了 lambda，发现它是真的方便。

当然这一题的难点就是排序的稳定性，但是只要采用稳定排序算法即可，例如插入，归并。不要用快排即可。