

暨南大学本科实验报告专用纸

课程名称 算法分析与设计实验 成绩评定
实验项目名称 最优服务安排问题 指导教师 李展
实验项目编号 实验 X 实验项目类型 综合性 实验地点
学生姓名 张印祺 学号 2018051948
学院 信息科学技术 系 计算机科学 专业 网络工程
实验时间 2020 年 4 月 8 日 ~ 4 月 8 日 下午 温度 °C 湿度

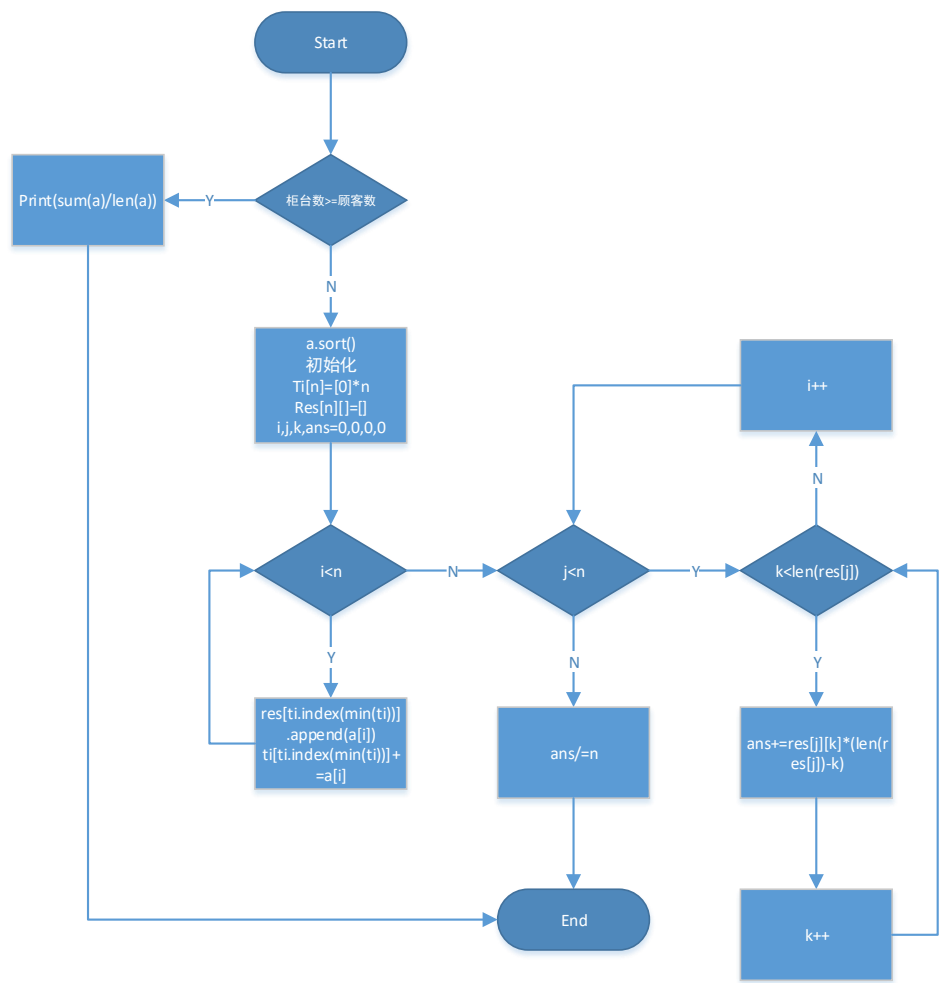
一、问题描述

设有 n 个顾客同时等待一个服务, 顾客 i 需要的服务时间为 $t(i)$, $1 \leq t(i) \leq n$ 。
共有 s 处可提供服务, 应如何安排顾客的服务次序才能使得平均等待时间达到最小?

二、算法思路

- 1、判断顾客数与柜台数关系, 若柜台数大于顾客数则无需等待, 并结束运行, 否则进入 2;
- 2、将顾客服务时间按升序排序;
- 3、用一个列表来记录每个柜台当前等候时间;
- 4、将当前等候队列顾客第一位安排至当前服务等候时间最短柜台;
- 5、将所有的等候时间求和取算术平均。

三、流程图



四、测试结果

测试案例 1.

```
a=[56,12,1,99,1000,234,33,55,99,812]
n=2
b=custom(a,n)

[56, 12, 1, 99, 1000, 234, 33, 55, 99, 812]
336.0
[[1, 33, 56, 99, 812], [12, 55, 99, 234, 1000]]
```

测试案例 2.

```
a=[56,12,1,99,1000,234,33,55,99,812]
n=11
b=custom(a,n)

[56, 12, 1, 99, 1000, 234, 33, 55, 99, 812]
240.1
无需等待
```

五、实验总结

要让顾客的平均等待时间最短，首先就要让服务总时间最短，安排使得服务时间最短的有效方法是利用贪心算法的多机调度算法。

对于一个队列中总的排队时间，算法如下：

队首的顾客等待时间为 0，之后每个顾客的等待时间为前一顾客的等待时间加上前一顾客的服务时间，以此类推。顾客的服务时间会在后面的顾客的等待时间中累加，因此要尽量安排服务时间短的顾客排在前面。

当顾客数量不大于服务点时，算法用时需要 $O(1)$ 时间。

当顾客数量大于服务点时，排序耗时 $O(n\log n)$ ，为 n 位顾客分配耗时 $O(n)$ 。因此算法总的计算时间为：

$$O(n\log n + n) = O(n\log n)。$$

证明此方法可以得出最优解：

假设原问题为 T （假设只有一个服务点），而我们已经知道了某个最优服务队列，即最优解为： $A=\{t(1), t(2), \dots, t(n)\}$ ，

则每个用户等待时间为：

$$T(1)=t(1); T(2)=t(1)+t(2); \dots T(n)=t(1)+t(2)+t(3)+\dots+t(n);$$

反证法来证明：

假设 $t(1)$ 不是最小的，不妨设 $t(i) > t(1) \quad i > 1$ 。

设另一服务序列 $B = (t(i), t(2), \dots, t(1), \dots, t(n))$

$$\text{那么 } T(A)-T(B)=n*[t(1)-t(i)]+(n+1-i)*[t(i)-t(1)]=(1-i)*[t(i)-t(1)]>0$$

即 $T_A > T_B$ ，这与 A 是最优服务相矛盾。

而我们可以非常轻易的看出 1 个柜台的安排其实是 n 个柜台安排的子问题。所以本题得出的是最优解。

六、附录

```
class custom:
    def __init__(self, a:list, n:int):
        print(a)
        if n>=len(a):
            print(sum(a)/len(a))
            print("无需等待")
            return None
        a.sort()
        self.Assign(a,n)
    def Assign(self, a:list,n:int):
        res=[]
        for i in range(n):
            res.append([0]*n)
        ti=[0]*n
        for i in range(len(a)):
            res[ti.index(min(ti))].append(a[i])
            ti[ti.index(min(ti))]+=a[i]
        ans=0
        for i in range(len(res)):
            for j in range(len(res[i])):
                ans+=res[i][j]*(len(res[i])-j)/len(a)
        print(ans)

a=[]
x=eval(input("请输入顾客人数: "))
for i in range(x):
    a.append(eval(input("请输入服务时长: ")))
n=eval(input("请输入柜台数: "))
b=custom(a,n)
```