

# 华东师范大学数据科学与工程学院上机实践报告

课程名称：算法设计与分析

年级：19 级

上机实践成绩：

指导教师：金澈清

姓名：龚敬洋

上机实践名称：贪心算法

学号：

上机实践日期：

10195501436

2020/12/11

上机实践编号：No.10

组号：1-436

## 一、目的

1. 熟悉算法设计的基本思想
2. 掌握贪心算法的思路

## 二、内容与设计思想

### 2.1 最优打印等待问题

某个班级有  $m$  位同学一起去打印店打印实验报告。这个打印店只有一台打印机。由于各实验报告的厚薄不同，所需打印时间也不相同。同学  $i$  需要的打印时间为  $t_i (1 \leq i \leq m)$ 。应该如何安排这些同学的打印次序使得平均等待时间最小？

输入数据格式：在文件 `students.txt` 中。第 1 行是正整数  $m$ ，表示有  $m$  位同学。接下来的一行中有  $m$  个正整数，表示  $m$  个同学所需要的打印时间。

输出数据格式：将最小平均等待时间输出到文件 `waittime.txt`。

### 2.2 零钱找零问题

张阿姨是超市售货员，经常需要给客户找零。为了买卖便利，张阿姨在找零钱的过程中通常返回最少张（枚）数零钱，市面上流通的钱币有 100, 50, 20, 10, 5, 1, 0.5, 0.1 元等各种样式，若某顾客购物消费  $m$  元，付款  $\lceil 1.05m \times 10 \rceil / 10$ ，则找零钱使用的最少钱币数量是：

输入数据格式：在文件 `customer.txt` 中。第 1 行是正整数  $a$ ，表示顾客有几次消费。第 2 行有  $a$  个整数，表示这些顾客的消费 1。

输出数据格式：将找零策略输出到文件 `output.txt` 中。共有  $a$  行。每行 8 个数字，分别表示不同面值钱币的数量。

## 三、使用环境

推荐使用 C/C++ 集成编译环境。

## 四、实验过程

### 1. 编写相关实验代码

最优打印等待问题：

```
1. #include <fstream>
2. #include <algorithm>
```

```

3. using namespace std;
4. int a[1000005];
5. int main() {
6.     ifstream fin("students.txt");
7.     ofstream fout("waittime.txt");
8.     int m, sum, pre;
9.     double mean;
10.    fin>>m;
11.    for(int i = 0; i < m; i++) fin>>a[i];
12.    sort(a, a + m);
13.    sum = 0;
14.    pre = 0;
15.    for(int i = 0; i < m - 1; i++){
16.        pre += a[i];
17.        sum += pre;
18.    }
19.    mean = (double)sum / m;
20.    fout<<mean;
21.    fin.close();
22.    fout.close();
23.    return 0;
24. }

```

零钱找零问题：

```

1. #include<fstream>
2. #include<cmath>
3. #include<cstring>
4. using namespace std;
5. int m[1000005];
6. int main(){
7.     ifstream fin("customer.txt");
8.     ofstream fout("output.txt");
9.     int a, rec[8];
10.    double nom[8] = {100, 50, 20, 10, 5, 1, 0.5, 0.1};
11.    double res;
12.    fin>>a;
13.    for(int i = 0; i < a; i++){
14.        fin>>m[i];
15.        res = floor(1.05 * m[i] * 10) / 10;
16.        memset(rec, 0, sizeof(rec));
17.        for(int j = 0; j < 8; j++){
18.            if(res / nom[j] >= 1){
19.                rec[j] = floor(res / nom[j]);
20.                res -= nom[j] * rec[j];
21.            }
22.            if(res < 0.1) break;
23.        }
24.        for(int w = 0; w < 8; w++) fout<<rec[w]<<" ";
25.        fout<<endl;
26.    }
27.    fin.close();
28.    fout.close();
29.    return 0;
30. }

```

2. 写出算法的思路，可能的话证明所设计的算法是否最优。

### 最优打印等待问题

算法思路：

将打印时间从小到大排序，打印时间小的先开始打印。

正确性证明：

设排好序的打印时间序列为  $\langle a_1, a_2, \dots, a_n \rangle$  ( $a_1 \leq a_2 \leq \dots \leq a_n$ )

则所有人的总等待时间

$$T = \sum_{k=1}^n \sum_{i=0}^{k-1} a_i$$

由于平均等待时间  $\bar{T} = \frac{T}{n}$ ，故要证  $\bar{T}$  最小，即要证  $T$  最小

下证该打印序列的总等待时间最小

任取  $p \neq q \in \{1, 2, \dots, n\}$ ，将  $a_p$  与  $a_q$  互换

不妨设  $p < q$ ，则此时所有人的总等待时间

$$\begin{aligned} T' &= \sum_{k=1}^p \sum_{i=0}^{k-1} a_i + \sum_{k=p+1}^q \left( \sum_{i=0}^{p-1} a_i + a_q + \sum_{i=p+1}^{k-1} a_i \right) + \sum_{k=q+1}^n \sum_{i=0}^{k-1} a_i \\ &= \sum_{k=1}^p \sum_{i=0}^{k-1} a_i + \sum_{k=q+1}^n \sum_{i=0}^{k-1} a_i + \sum_{k=p+1}^q \sum_{i=0}^{p-1} a_i + \sum_{k=p+1}^q \sum_{i=p+1}^{k-1} a_i + (q-p)a_q \end{aligned}$$

而

$$T = \sum_{k=1}^p \sum_{i=0}^{k-1} a_i + \sum_{k=q+1}^n \sum_{i=0}^{k-1} a_i + \sum_{k=p+1}^q \sum_{i=0}^{p-1} a_i + \sum_{k=p+1}^q \sum_{i=p+1}^{k-1} a_i + (q-p)a_p$$

又由于  $a_p \leq a_q$ ，故  $T' \geq T$

因此互换序列中任意两人的打印顺序所得的总等待时间均不小于原序列的总等待时间

即原序列的总等待时间最小

## 零钱找零问题

算法思路：

将纸币的面值从大到小排序，当大的纸币单张面额超过剩下的找零时再用更小的纸币继续找零

正确性证明：

设钱币面额分别为  $\langle a_1, a_2, \dots, a_8 \rangle$  ( $a_1 > a_2 > \dots > a_8$ )，需要找零的数额  $r = \frac{[1.05m \times 10]}{10} - m$ ，贪心算法下对应的找零钱币个数为  $\langle k_1, k_2, \dots, k_8 \rangle$

则

$$r = \sum_{i=1}^8 k_i a_i$$

下证明  $s = \sum_{i=1}^8 k_i$  最小

若在某一种面额的钱币上少用一张换作更小面额的钱币，即  $k'_p = k_p - 1$  ( $p \in \{1, 2, \dots, 8\}$ )

则

$$r = \sum_{i=1}^{p-1} k_i a_i + k_p a_p - a_p + \sum_{i=p+1}^8 k'_i a_i$$

由于  $a_i < a_p$  ( $i > p$ )，故  $\sum_{i=p+1}^8 (k'_i - k_i) > 1$

故  $s' = \sum_{i=1}^8 k'_i > s$

故使用贪心算法生成的找零钱币个数总和最小

3. 把各组实验的输入和输出文件打印出来。

#### **最优打印等待问题**

输入: (students.txt)

10

8 7 5 2 19 6 20 8 1 15

输出: (waittime.txt)

23.6

#### **零钱找零问题**

输入: (customer.txt)

8

111 222 333 567 234 256 666 789

输出: (output.txt)

0 0 0 0 1 0 1 0

0 0 0 1 0 1 0 0

0 0 0 1 1 1 1 1

0 0 1 0 1 3 0 2

0 0 0 1 0 1 1 1

0 0 0 1 0 2 1 3

0 0 1 1 0 3 0 2

0 0 1 1 1 4 0 3

### **五、总结**

对上机实践结果进行分析, 问题回答, 上机的心得体会及改进意见。

通过实验发现, 最优打印等待问题和零钱找零问题均可以使用贪心算法得到正确的结果, 文中也从数学上严格证明了贪心算法的正确性。相比动态规划算法, 使用贪心算法解决这两个问题的运行效率更高, 可以更快的求解出所需的结果。