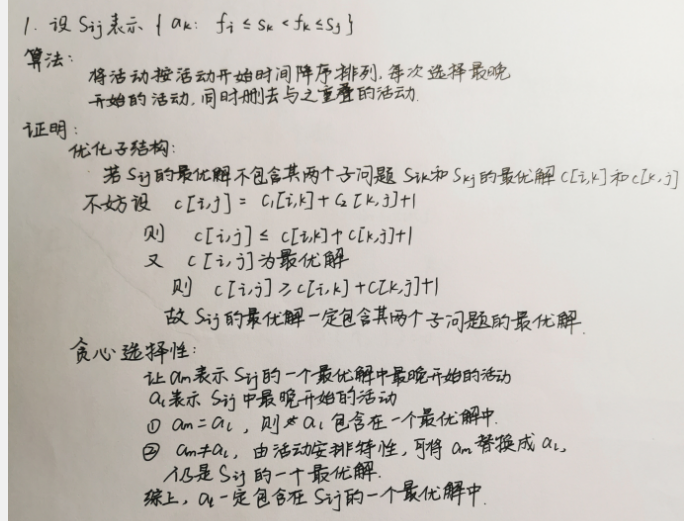
**算法设计与分析第五章作业**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **姓名** |  | **班级** |  | **学号** |  |
| **第1题** |  | | | | |
| **第2题** |  | | | | |
| **第3题** |  | | | | |
| **第4题** |  | | | | |
| **第5题** |  | | | | |
| **总分** |  | | | | |
| **备注** | 作业提交截止时间：2020-10-14日24:00，超过提交截至时间的作业视为无效。确因网络等特殊原因无法及时提交作业的学生，应至少提前1小时与助教。作业文件名命名方式： 第x章-x班-姓名-学号（例， 第1章-1班-张三-2018054000.doc）； 邮件主题为：第x章作业, x班，姓名，学号（例， 第1章作业，1班，张三，2018054000）。缺少这些信息的作业将被酌情扣分。 | | | | |

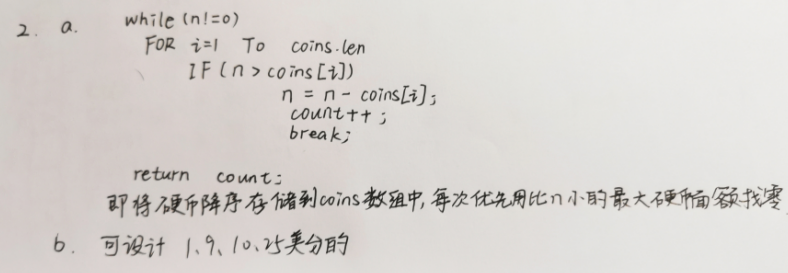
**作业请严格按照要求提交，否则将酌情扣分！**

1. **（30分）**假定我们不再一直选择最早结束的活动，而是选择最晚开始的活动，前提仍然时与之前选出的所有活动兼容。描述如何利用这一方法设计贪心算法，并证明算法会产生最优解。
2. 

**2、（30分）**考虑用最少的硬币找n美分零钱的问题。假定每种硬币的面额都是整数。

a.设计贪心算法求解找零问题。假定有25美分、10美分、5美分和1美分4种面额的硬币。

b.设计一组硬币面额，使得贪心算法不能保证的到最优解。这组硬币面额中应该包含1美分，使得对每个零钱值都存在找零方案。



**3、（40分）编程题：**柠檬水找零

题目描述：

在柠檬水摊上，每一杯柠檬水的售价为 5 美元。

顾客排队购买你的产品，（按账单 bills 支付的顺序）一次购买一杯。

每位顾客只买一杯柠檬水，然后向你付 5 美元、10 美元或 20 美元。你必须给每个顾客正确找零，也就是说净交易是每位顾客向你支付 5 美元。

如果你能给每位顾客正确找零，返回 true ，否则返回 false 。

提示：

0 <= bills.length <= 10000

bills[i] 不是 5 就是 10 或是 20

示例 1：

输入：[5,5,5,10,20]

输出：true

解释：

前 3 位顾客那里，我们按顺序收取 3 张 5 美元的钞票。

第 4 位顾客那里，我们收取一张 10 美元的钞票，并返还 5 美元。

第 5 位顾客那里，我们找还一张 10 美元的钞票和一张 5 美元的钞票。

由于所有客户都得到了正确的找零，所以我们输出 true。

示例 2：

输入：[5,5,10]

输出：true

示例 3：

输入：[10,10]

输出：false

示例 4：

输入：[5,5,10,10,20]

输出：false

解释：

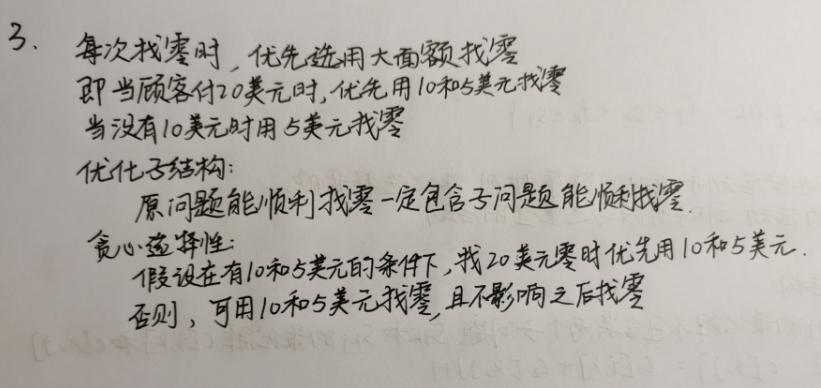
前 2 位顾客那里，我们按顺序收取 2 张 5 美元的钞票。

对于接下来的 2 位顾客，我们收取一张 10 美元的钞票，然后返还 5 美元。

对于最后一位顾客，我们无法退回 15 美元，因为我们现在只有两张 10 美元的钞票。

由于不是每位顾客都得到了正确的找零，所以答案是 false。

**要求：运用贪心思想作答，请写出分析过程，并用一种语言（最好是C++或JAVA）实现你的思路，上交作业时请将代码一并提交，代码粘贴在交作业的word里面，复杂度尽可能低。**



Scanner sc = **new** Scanner(System.***in***);  
ArrayList<Integer> billList = **new** ArrayList();  
**int**[] store ={0,0,0};  
**boolean** flag = **true**;  
**int** money;  
**while**(!sc.hasNext(**"#"**)){  
 money = sc.nextInt();  
 **if**(money == 5){  
 billList.add(money);  
 store[0]++;  
 }  
 **else if**(money == 10){  
 **if**(store[0]>=1) {  
 store[0]--;  
 store[1]++;  
 }  
 **else** flag = **false**;  
 }  
 **else if**(money == 20){  
 **if**(store[1] >= 1 && store[0] >= 1){  
 store[2]++;  
 store[1]--;  
 store[0]--;  
 }  
 **else if**(store[0] >= 3){  
 store[2]++;  
 store[1] -= 3;  
 }  
 **else** flag = **false**;  
 }  
}  
return flag;