

# 算法设计与分析第 5 章

毛翰翔

210110531

2022 年 10 月 17 日

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，前提仍然是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。

**Solution:** 类似地，我们将活动按照开始时间从小到大进行排列，并从最后一个活动开始选择。剩下的引理其实完全类似。记号沿用课内介绍的。设  $S = \{A_i : A_i = (s_i, f_i), s_i \leq s_j \text{ if } i < j\}$ ,  $A$  for activity,  $s$  for start,  $f$  for fin.

优化解一定包括了最后一个活动，记为  $A_n$ ，对于一个没有最后一个活动的优化解，我们将其最晚开始的活动换为  $A_n$ ，这个解仍是优化解。

给定一个包含了最后一个活动的优化解，将最后这个活动去掉，所剩下的正是  $S$  的子集  $S' = \{A_i : f_i < s_n\}$  的优化解。

这上面两点已经足够说明贪心选择性，这点只需要归纳法就足以证明，我们每一次加进来的活动都满足：存在优化解包含了它。那么到最后我们所加入的所有活动构成了优化解。

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。

**Solution:**

a: 对于给定的  $n$  美分，我们从最大面额的硬币开始选择，如果说  $n \geq 25$  那么我们就找  $\lfloor n/25 \rfloor$  枚 25 美分的硬币。对于剩下的  $n \bmod 25$ ，类似地，能够选取多少大面额的硬币就选取多少，以此类推。因为能够选择 1 美分的硬币，所以一定能够找零。

对于找零问题的一个优化解，如果  $n \geq 25$  则，25 美分的硬币的数量一定是  $\lfloor n/25 \rfloor$ 。因为 25 能够被 10, 5 找零。

b: 1, 5, 8

不能通过贪心算法得到最优解。比如说给定 10，按照贪心算法，解应该是 8, 1, 1，但是最优解是 5, 5

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。注意，一开始你手头没有任何零钱。如果你能给每位顾客正确找零，返回 true，否则返回 false。提示:  $0 \leq \text{bills.length} \leq 10000$  bills[i] 不是 5 就是 10 或是 20

---

```
1  #include <stdio.h>
2  int coins (){
3      int five = 0, ten = 0;
4      // initialization
5      int n = 0, current;
6      // n for the length of bills
7      scanf("%d", &n);
8      for (int i = 0 ; i < n ; i++){
9          scanf("%d", &current);
10         switch (current){
11             case 5:
12                 five++; break;
13             case 10:
14                 ten++; five = five - 2;
15                 if (five < 0)
16                     return 0 ;
17                 break;
18             case 20:
19                 if (five - 3 < 0 && (ten - 1 < 0 || five - 1 < 0))
20                     return 0 ;
21                 else if (ten - 1 < 0 || five - 1 < 0)
22                     five = five - 3;
23                 else {
24                     five--;
25                     ten--;
26                 }
27                 break;
28             default:
29                 break;
30         }
31     }
32     return 1;
33 }
34
35 int main (){
36     int ans = coins();
37     if (ans)
38         printf("true\n");
39     else
40         printf("false\n");
41     return 0;
42 }
```

---