

Python程序设计

找零钱与简单贪心

苏州大学计算机科学与技术学院

找零钱

- Jerry带了1美元去超市买糖果
 - Jerry买了34美分的糖果
 - 超市只有4种找零硬币
 - 25美分、10美分、5美分和1美分
 - Jerry付钱的时候希望找零硬币的个数最少
 - 应该如何找零?
- $25+25+10+5+1=66$



2



Quarter

1



Dime

1



Nickel

1



Penny

贪心法

- 又称为贪婪法
- 该方法在对问题求解时，总是做出当前最好的选择
- 具体思路：
 1. 根据局部最优原则，把待求解问题分成若干个子问题
 2. 对每一个子问题求解，得到子问题的解
 3. 把子问题的局部最优解合并得到整个问题的解

找零钱的贪心处理过程

- 找零钱66美分，有4种硬币，可以分解成4个子问题
 1. 计算出最多使用几个25美分，得到剩余的金额
 2. 计算出最多使用几个10美分，得到剩余的金额
 3. 计算出最多使用几个5美分，得到剩余的金额
 4. 计算出最多使用几个1美分
- 合并上述4步的结果得到问题的解
- 上述四个子问题的先后顺序不可错位

找零钱的代码

```
money=66                #初始金额

quarters=money//25      #子问题1, 计算25美分硬币个数
money=money%25          #得到剩余金额

dimes=money//10         #子问题2, 计算10美分硬币个数
money=money%10         #得到剩余金额

nickels=money//5        #子问题3, 计算5美分硬币个数
pennys=money%5         #子问题4, 计算1美分硬币个数

print("找零25美分硬币{0}个".format(quarters))
print("找零10美分硬币{0}个".format(dimes))
print("找零5美分硬币{0}个".format(nickels))
print("找零1美分硬币{0}个".format(pennys))
```

找零25美分硬币2个
找零10美分硬币1个
找零5美分硬币1个
找零1美分硬币1个

运行结果

用循环实现的找零代码

```
money=int(input("请输入找零金额（1-99）：")) #初始金额

coins=(25, 10, 5, 1)
counts=[]

for coin in coins:
    counts.append(money//coin)
    money%=coin

for i in range(len(counts)):
    print("找零{0}美分硬币{1}个".format(coins[i], counts[i]))
```

贪心的缺陷

- 小明带着一个背包来到免税商店
 - 背包最大负重为3.5公斤
 - 选择要求
 - 装入的物品的总价格最大
 - 不超出背包最大重量

商品	重量	价格
二胡	1.5	1500
音响	3	3000
笔记本电脑	2	2000
投影仪	2.9	2999
钢笔	0.1	200

- 以重量为贪心条件，结果：音响+钢笔=3200
- 以价格为贪心条件，结果：音响+钢笔=3200
- 以价格/重量为贪心条件，结果：笔+投影仪=3199
- 二胡+笔记本电脑=3500

小结

- 贪心过程：
 - 自顶向下
 - 基于步步最优的原则
 - 得到当前步的最优解，并逐步把原问题分解为更加小的问题
 - 最后合并每一步的解
- 优点
 - 简单、高效
- 缺点
 - 不能保证找到最优解

思考题

- 某奶茶店促销
 - 奶茶10元一杯；
 - 每买3杯送1杯或每买5杯送2杯；
 - 小强带了N元钱 ($N \geq 0$)；
 - 请问小强最多可以买多少杯？

