

《枚举算法（二）》学程拓展

（一）银行卡的密码为什么输错 3 次，卡就被锁定了？

我国的银行卡都限制了密码输入次数，如果密码输错 3 次，账户就会被锁定。这是为什么呢？

这样做主要是为了避免别有用心的人使用枚举算法暴力破解密码。由于银行卡密码是在键盘数字键所限定的范围内设置。枚举算法的特点之一是逐个检验枚举对象的每个解，所得到的结果肯定是正确的。也就是将所有可能的密码组合一个一个去尝试最终破解银行卡的密码。为了应对别有用心的人利用枚举算法编写程序盗刷银行卡这种情况，多数登录系统会对密码输入次数进行限制。当错误次数达到可容许次数时，密码验证系统就会自动拒绝验证，甚至还会自动启动入侵警报。

（二）如何对“硬币方案”问题进一步优化，去掉枚举对象 C？

优化分析：

- 利用约束条件：“总数量 50 个”和公式推导进行优化

（一）由题面可知：

$$\begin{cases} a+b+c+d=50 & \text{(等式 1)} \\ a+b/2+c/10+d/20=20 & \text{(等式 2)} \end{cases}$$

（二）由等式 2 可推出： $20a+10b+2c+d=400$ （等式 3）

（三）由等式 3 减 1 等式 1 可推出： $19a+9b+c=350$

所以： $c=350-19a-9b$ （c 可用 a, b 表示，即可去除枚举对象 c）

（四）d 可表示为： $50-a-b-c$ （d 可用 a, b, c 表示，即可去除枚举对象 d）

- 利用约束条件：“总金额 20 元”和边界条件构建判定条件

判定条件表达式： $a*1+b*0.5+c*0.1+d*0.05==20$ and $d \geq 0$ and $c \geq 0$

- 综上分析，可得出优化程序（参考）：

```
times=0
num=0
for a in range(0, 21):
    for b in range(0, 51-a):
        times=times+1
        c=350-9*b-19*a
        d=50-a-b-c
        if (a*1+b*0.5+c*0.1+d*0.05==20 and d>=0 and c>=0):
            num=num+1
print("共有", num, "种方案", "总次数", times)
```

- 运行结果：

```
=====
共有 50 种方案 枚举次数 861
>>>
```