



数据结构与算法 (Python版)

优化问题和贪心策略

陈斌 北京大学 gischen@pku.edu.cn

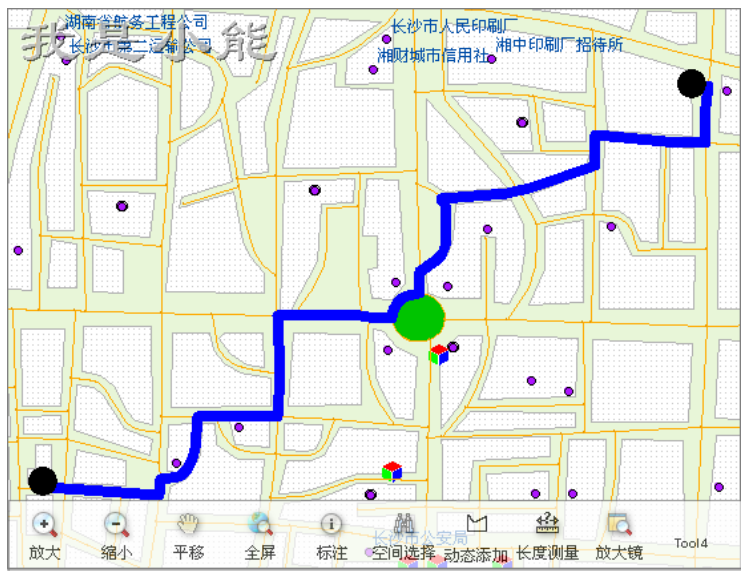
优化问题

❖ 计算机科学中许多算法都是为了找到某些问题的**最优解**

例如，两个点之间的最短路径；

能最好匹配一系列点的直线；

或者满足一定条件的最小集合



找零兑换问题

❖ 一个经典案例是兑换最少个数的硬币问题

假设你为一家自动售货机厂家编程序，自动售货机要每次找给顾客最少数量硬币；

假设某次顾客投进\$1纸币，买了¢37的东西，要找¢63，那么最少数量就是：2个quarter (¢25)、1个dime (¢10) 和3个penny (¢1)，一共6个



贪心策略解决找零兑换问题

❖ 人们会采用各种策略来解决这些问题，例如最直观的“**贪心策略**”

❖ 一般我们这么做：

从最大面值的硬币开始，用尽量多的数量

有余额的，再到下一最大面值的硬币，还用尽量多的数量，一直到penny (1¢) 为止



贪心策略Greedy Method

❖ 贪心策略

因为我们每次都试图解决问题的**尽量大的一部分**
对应到兑换硬币问题，就是**每次以最多数量的最大面值硬币来迅速减少找零面值**

❖ “贪心策略” 解决找零兑换问题，在美元或其他货币的硬币体系下表现尚好



贪心策略失效

❖ 但如果你的老板决定把自动售货机出口到 Elbonia, 事情就会有点复杂

(系列漫画Dilbert里杜撰的国家)

因为这个古怪的国家除了上面3种面值之外, 还有一种【 $\epsilon 21$ 】的硬币!



贪心策略失效

- ❖ 按照“贪心策略”，在Elbonia， $\text{¢}63$ 还是原来的6个硬币

$$\text{¢}63 = \text{¢}25*2 + \text{¢}10*1 + \text{¢}1*3$$

- ❖ 但实际上最优解是3个面值 $\text{¢}21$ 的硬币！

$$\text{¢}63 = \text{¢}21*3$$

- ❖ “贪心策略”失效了

Inflation in Elbonia



PROBLEM SOLVED