武松打虎

西南大学附属中学
High School Affiliated to Southwest University

问题描述:

曾经因打虎而文明的武松在x年后接到了景阳冈动物园的求助信，信上说:最近我们动物园逃跑了几只老虎，请您把它们抓回来，thank you!武松接到信后立刻上了山。

正当他到半山腰是，suddenly! 跳出n只猛虎来。每只老虎都有一块虎牌，牌上写着的是每一只虎最大拥有的体力，当武松与老虎PK时，若老虎的体力先用完，那么老虎over,否则武松over。求武松在over之前最多能干掉几只老虎?

(注:老虎是一只只上的)

输入:
第一行两个数字n(n<50000), t0(武松的体力)。第二行n个数字，分别代表每只老虎的体力。所有变量都不超过long int范围。
输出:
一行，最多能干掉的老虎数。

样例输入:
6 10
1 5 3 2 4 6
样例输出:
4

Q：如何打虎，才能干掉最多的老虎?

1.以我们考试解题为例：

在考试时，如果每道题分值相同的话，时间一定的情况下为了得到更高的分数我们会倾向于先做简单题再做难题，因为难题比简单题费时更长、更难得分。

2.那么假设打死一只老虎得分为1，它的体力表示难易程度，那么我们很容易得到这样的策略：

每次打体力最小的老虎 → 武松能剩余更多体力，打更多的老虎

贪心算法

在现实生活中，我们常常会想着以目前看起来“最好，最优，最有效率”的方式来解决问题，其实这就是贪心的体现。

信息学

贪心算法

西南大学附属中学校

信息奥赛教练组

基本思想：

将问题的求解过程看作是一系列选择，每次选择都是**当前状态下的最好选择(局部最优解)**，不断地做出选择，直到问题求解完成。

以武松打虎为例：

1.t0=10, 干掉体力为1 5 3 2 4 6的老虎 **选择干掉1**

↓ 问题转化为

2.t0=9，干掉体力为5 3 2 4 6的老虎 **选择干掉2**

↓ 问题转化为

3.t0=7，干掉体力为5 3 4 6的老虎 **选择干掉3**

↓ 问题转化为

4.t0=4，干掉体力为5 4 6的老虎 **选择干掉4**

↓ 问题转化为

5.t0=0，干掉体力为5 6的老虎 **over**

每次都是当前问题最好的选择，即打死体力值最小的老虎

贪心选择前，先排序

贪心解题需满足的两个性质：

1.最优子结构性质

2.贪心选择性质

1.最优子结构性质

当一个问题的最优解包含其子问题的最优解时，称此问题具有最优子结构性质。

Q：什么是子问题?

1.t0=10, 干掉体力为1 5 3 2 4 6的老虎 **选择干掉1**

↓ 问题转化为

2.t0=9，干掉体力为5 3 2 4 6的老虎 **选择干掉2**

↓ 问题转化为

3.t0=7，干掉体力为5 3 4 6的老虎 **选择干掉3**

↓ 问题转化为

4.t0=4，干掉体力为5 4 6的老虎 **选择干掉4**

通过每次最优的选择，问题慢慢变成一个**问题描述和所求答案与原问题相同但数据规模更小**的问题，称为**子问题**

1.最优子结构性质

问题1.t0=10, 干掉体力为1 5 3 2 4 6的老虎

子问题

问题2.t0=9，干掉体力为5 3 2 4 6的老虎

答案的解为：1 2 3 4

答案的解为：2 3 4

左边大问题的解包含右边子问题的解

2.贪心选择性质

所求问题的**整体最优解可以通过一系列局部最优的选择得到**，也就是说，当前的选择只依赖于之前的选择，不依赖于后面的选择

武松打虎的贪心策略：每次都打死体力值最小的老虎

我们知道，武松剩余的体力越多能够打死的老虎就越多。
如果我们采用上述的贪心方式，能够保证每次打死老虎后武松剩余的体力最大，最终能保证武松打死的老虎最多。
即通过一系列局部最优解得到整体最优解，这种贪心策略满足贪心选择性质。

贪心选择性质的关键是选取怎样的贪心策略

疑问 贪心算法如何保证每一步的最优解最终能逐步达到整体的最优解呢？

需要进行正确性证明

贪心算法求解的一般步骤：

1.通过问题信息找规律、分析归纳(**一般要结合排序找规律**)

2.进行贪心猜想

3.对猜想进行正确性证明(严格证明和一般证明)

(1) 严格证明：数学归纳法、反证法、矩阵胚理论

(2) 一般证明：列举反例即可

4.设计并实现程序

经典应用：活动安排、乘船问题、区间问题等

例1：独木舟的旅行

进行一次独木舟的旅行活动，独木舟可以在港口租到，并且之间没有区别。一条独木舟最多只能乘坐两个人，且乘客的总重量不能超过独木舟的最大承载量。我们要尽量减少这次活动中的花销，所以要找出可以安置所有旅客的最少的独木舟条数。现在请写一个程序，读入独木舟的最大承载量、旅客数目和每位旅客的重量。根据给出的规则，计算要安置所有旅客必须的最少的独木舟条数，并输出结果。

输入
第一行输入s,表示测试数据的组数;
每组数据的第一行包括两个整数w，n，
80<=w<=200,1<=n<=300，w为一条独木舟的最大承载量,n为人数;
接下来的一组数据为每个人的重量（不能大于船的承载量）；
输出
每组人数所需要的最少独木舟的条数。

样例输入
2
85 6
5 84 85 80 84 83
90 3
90 45 60
样例输出
5
3

题意

每条船的承载量为C,怎么安排让船的数量最少

Q：如何才能使船的数目最少？

每艘船都尽可能达到最大承载量C

策略：
1）将所有人的重量w进行排序；
2）从当前最轻的重量w_i开始考虑，找能跟其坐一只船的最重的人w_j
3）比最重的人j都重的人都单独坐一个船；

正确性证明

反证法：

1.假设i不与任何人同船，如果将j拉来与其同船，用的船数会小于等于原来的船数；
2.假设i与k同船，因为j是与i匹配的最重的，所以w(k)<=w(j)，则j加入其它船可能会使其它船超重，用的船数会变多；
综上，说明这样的贪心法不会丢失最优解。

例二、最大整数

设有n个正整数（n<=20），将它们连接成一排，组成一个最大的多位整数。
例如：n=3时，3个整数13,312,343,连接成的最大整数为：34331213
又如：n=4时，4个整数7,13,4,246连接成的最大整数为：7424613

输入
第一行：正整数n（n<=20）
第二行：n个整数，两个数之间用一个空格隔开
输出
连接成的多位数

样例输入
3
13 312 343
样例输出
34331213

目的：使得连接起来的数字所代表的整数最大

通过分析样例猜想：

按数字的字典序比较，342>312>13，最终结果是按照字典序从大到小进行连接。

正确性证明：

反例：

例如A=“12”，B=“121”，121>12，而明显12121>12112,即AB>BA。

进一步思考：

可以发现，字符串A和B的大小关系并不是他们本身来决定，
而是他们组合后的数来决定。
这里应该**判断AB和BA的大小关系**来判断A和B的大小关系：
若是AB>BA则A>B，若是AB<BA则A<B，否则A==B。

最终的贪心策略：按照上述的比较规则对所有字符串从大到小排序，依次输出即可。

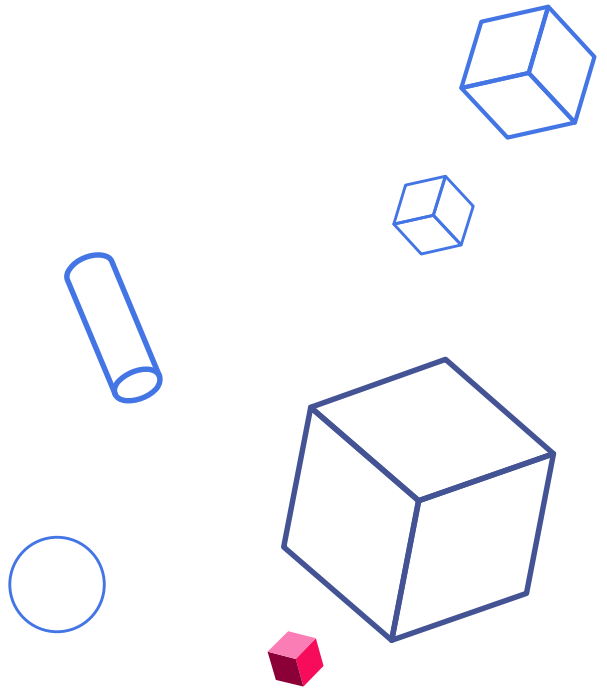
算法评价

贪心算法在解决问题的策略上**目光短浅**，只根据当前已有的信息就做出选择，而且一旦做出了选择，不管将来有什么结果，这个选择都不会改变。
换言之，贪心法并不是从整体最优考虑，它所做出的选择只是在某种意义上的局部最优。

如果贪心算法被证明是不正确的，最后得到的解只能是一个**较优解**

FZOI
西大附中信息学竞赛
High School Affiliated to Southwest University

明天：贪心经典模型-区间问题



Thanks

For Your Watching