友好的动物解题报告

广东中山纪念中学 陈启峰

【问题描述】

W 星球是一个和地球一样气候适宜、物种聚集的星球。经过多年的研究，外星生物学家们已经发现了数万种生物，而且这个数字还在不断增大。

W 星球上的生物很有趣，有些生物之间很友好，朝夕相伴，形影不离；但有些却很敌对，一见面就难免发生战斗。为了能够更好地了解它们之间的友好程度，外星生物学家希望进行一些量化的计算。他们发现，两种生物之间的友好程度和它们的 $k(k \le 5)$ 种属性有关，暂且将它们编号为属性 1、属性 2、……、属性 k，这些属性都是可以进行量化的。外星生物学家研究发现，如果前 k-1 种属性的差别越大，这两种生物就越友好；但是属性 k 与众不同，这种属性差别越小的两种生物越友好。

因此他们猜想是不是可以用这样一个公式量化两种生物之间的友好程度：

$Friendliness = (\sum_{i=1}^{k-1} C_i \times$属性i的差别$) - C_k \times$属性k的差别，其中 C_i 是非负常数。

如果知道了每种生物的各种属性，利用上述公式就很容易算出它们之间的友好程度了。现在，外星生物学家们想问一问：在目前发现的这些生物当中，关系最友好的那对生物是哪一对呢？它们之间的友好程度是多少？

【问题分析】

【数学模型】

首先，把题目的任务转化为简明的数学语言：

令 $A_{i,j} = C_i \times$第i个动物的属性j,求出两种动物的编号：

1<=a<b<=n

使得目标函数

$$Friendliness(a,b)=(\sum_{i=1}^{k-1}|A_{a,i}-A_{b,i}|)-|A_{a,k}-A_{b,k}| \quad ①$$

最大化。

【原始算法的矛盾】

显然最原始的算法是枚举所有可能的 a 和 b，求出最大的 *Friendliness*。但这样做程序的时间复杂度是 O（$n^2 \times k$），显然无法通过所有的数据。

【数据范围分析】

一个很显眼的条件 $k \le 5$，便令人不禁会想到时间复杂度为 O $(n \times 2^k)$、O（$n \times k!$），O（$n\log^k n$）之类的指数级算法,并促使人向该方面的算法思考。

【数据结构优化时的矛盾】

考虑从数据结构上优化原始算法。

首先，把函数①中的绝对值去掉，目标函数变成

$$Friendliness(a,b)=(\sum_{i=1}^{K-1}\pm A_{a,i}\mp A_{b,i})-(\pm A_{a,k}\mp A_{b,k}) \quad ②$$

其中小括号内前面是加号的数不小于前面是减号的数

这样就有 2^k 种情况需要考察。

然后，对于每一种动物 a,考察 a 的属性在函数②中前面的符号的情况，用 0 到 2^k -1 一个二进制 BN 表示正负情况：如果 BN 的第 i 位为 1，则 $A_{a,i}$ 前面的符号为正，令 SIGN(BN,i)=1,否则为负，令 SIGN(BN,i)= -1。定义

$$F_{a,BN}=(\sum_{i=1}^{k-1}A_{a,i}\times SIGN(BN,i))-A_{a,k}\times SIGN(BN,k)$$

表示正负情况为 BN 动物 a 的值总和。

则 Friendliness 的最大值为

$$\max\{F_{a,BN}+F_{b,2^k-1-BN}\} \quad ③$$

其中 a≠b,并且对于任意的 i∈[1,k],都有

$$A_{a,i} \times SIGN(BN,i) + A_{b,i} \times SIGN(2^k - 1 - BN,i) \geq 0 \qquad ④$$

实现时，只要先枚举 a 和 BN，然后查找满足性质④并且 $F_{b,2^k-1-BN}$ 值最大的 b(a<b)。这时的查找依懒于 2^k 棵第 k 层以 max{F}为元素、第 1 到第 k-1 层以线段树为元素的线段树。

复杂度分析：每次查找和插入的时间复杂度都为 O（$\log^k$ n），总的时间复杂度为 O ($n \times 2^k \times \log^k n$),空间复杂度为 O($n \log^{k-1} n$)，编程复杂度高。

此时矛盾重重，时间复杂度依然很高，空间复杂度太大，编程复杂度令人难以忍受。

【分析矛盾】

不难发现，产生这些矛盾的根本原因是性质④的要求太苛刻了。因此，此时须要放宽限制。

【“放宽”方法化解矛盾】

正所谓退一步海阔天空，试着放宽性质④的要求。

尝试将苛刻的要求去掉，但这样编出来的程序和用原始算法编出来的程序的运行结果是不正确的。因为这样做实在太武断了。

先来对性质④做个详细分析：注意到一个特殊性：$|A_{a,k} - A_{b,k}|$越大，Friendliness 的值越小，其余的$|A_{a,i} - A_{b,i}|$（i<k）越大，Friendliness 的值越小。因此，试着把性质④放宽为只要保证

$$A_{a,k} \times SIGN(BN,i) + A_{b,k} \times SIGN(2^k - 1 - BN,i) \geq 0 \qquad ⑤$$

结果，这样编出来的程序和用原始算法编出来的程序的运行结果是完全一样的。那么，这做法是否一定可以保证正确呢？下面的证明肯定了这一做法的正确性。

证明：

[定理]对于任意的 **A,B∈R,**都有**|A-B|≥A-B**，**|A-B|≥B-A**。

[推论]对于任意的 **A,B∈[1,n]**，**BN∈[0,2^k -1]**都有

$$(\sum_{i=1}^{K-1} |A_{a,i} - A_{b,i}|) - |A_{a,k} - A_{b,k}| \geq \text{满足性质⑤的} F_{a,BN} + F_{b,2^k-1-BN}$$

也就是

满足性质④的 $F_{a,BN1}+F_{b,2^k-1-BN1}$ ≥满足性质⑤的 $F_{a,BN2}+F_{b,2^k-1-BN2}$

设满足性质④的(a,b,BN)所构成的集合为 S1,满足性质⑤的所构成的集合为 S2，MAX1= $\max\{F_{a,BN}+F_{b,2^k-1-BN}\mid(a,b,BN)\in S1\}$

MAX2= $\max\{F_{a,BN}+F_{b,2^k-1-BN}\mid(a,b,BN)\in S2\}$

∵性质④包含性质⑤

∴S1⊆S2

∴MAX1≤MAX2

由推论可以得到：任意的（**a,b,BN2**）**∈S2** 对应的 $F_{a,BN2}+F_{b,2^k-1-BN2}$ ，都有一个**(a,b,BN1) ∈S1** 对应的 $F_{a,BN1}+F_{b,2^k-1-BN1}$ 大于或等于这个值。

∴MAX1≥MAX2

∴MAX1=MAX2

所以只要保证性质⑤，求出来的函数③的最大值就是 Friendliness 的最大值。

证毕。

实现时，为了保证满足性质⑤，只要先以 $A_{?,k}$ 为关键字将 A 从小到大排序，令 $Best_{i,BN}$ 为 $\max\{F_{a,BN}\mid a\le i\}$，这只要 O (n×$2^k$)的时间就可以把 $Best$ 计算出来 然后，枚举 i 和第 k 位为 1 的 BN，以 $Best_{i-1,2^k-1-BN}+F_{i,BN}$ 更新最大值。最后输出最大值。

复杂度分析：时间复杂度为 O(nlogn+n×2^k)。空间复杂度为 O(n×2^k)。

到此，利用“放宽”方法已经很好地解决了此题。

【小结】

归纳上面解题的步骤：可以得出用“放宽”方法解决最大化（最小化）的一种方法：

1、把任务转化成数学模型；
2、找出原始算法；
3、用数据结构优化原始算法；
4、分析优化后的苛刻条件，将其转化为宽松的条件。