从特殊情况考虑

复旦附中　李天翼

[关键字]　特殊情况　信息学竞赛

[摘要]

从特殊情况考虑是一种重要的数学思想。而特殊情况主要分为简单情况和极端情况。

本文通过几道例题，来说明从特殊情况考虑这一思想在信息学竞赛中的应用，并提炼出它们的共同点，揭示这一思想的重要内涵。

[目录]

总结

[正文]

例 1

1.问题描述（由 POI 2003-2004 Bra 改编）

考虑一个有 n 个门组成的电路。这些门被标号为 0、1、2、……、n-1。每个门有固定数目的输入和一个输出。输入和输出可以是 0、1、1/2 三种状态中的任意一个。每个输入连接某个门的一个输出。输入的状态与它所连接的输出状态相同。每个输出可以与数个输入相连。标号为 0 和 1 的门很特殊，它们没有输入，标号为 0 的门总输出 0，标号为 1 的门总输出 1。我们说，一个门的输出状态是“有效”的，当且仅当满足下列条件之一。

a)它等于 0 并且这个门的输入中 0 比 1 多。

b)它等于 1/2 并且这个门的输入中 0 和 1 一样多。

c)它等于 1 并且这个门的输入中 1 比 0 多。

d)它等于这个门的编号，且这个门的编号是 0 或 1。

如果所有的门的输出状态是“有效”的，那么我们说这个电路是“有效”的。如果一个门的输出状态在所有“有效”的电路中都是一样的，那么它的输出状态是固定的。保证存在“有效”的电路。

任务：

写一个程序

从标准输入中读取电路的描述

对每一个门，检查它的输出状态是否是固定的，如果是固定的，确定它的状态。

向标准输出中写入输出状态固定的门的状态

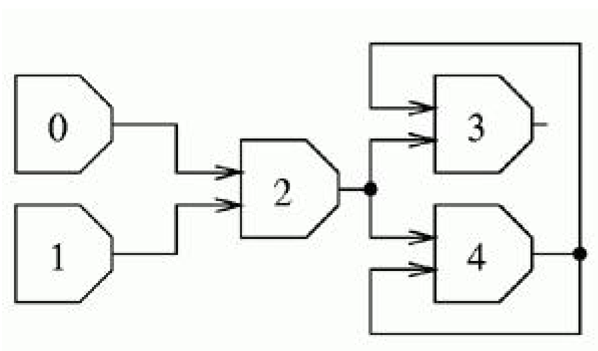
输入：

标准输入包含一个整数 n，$2 \le n \le 10000$。接下来的 n-2 行包括每个门的连接的描述。第 i 行描述第 i 个门的输入：第一个整数 k_i($k\_i \ge 1$)，表示这个门有 k_i 个输入，接下来的 k_i 个数表示这 k_i 个门的编号。行内整数之间用空格分隔。每个门的输入的总数不超过 200000。

输出：

你的程序应该输出 n 行到标准输出中。第 i 行包括的内容，取决于编号为 i-1 的门的输出状态。

- 0---如果它总是 0
- 1/2---如果它总是 1/2
- 1---如果它总是 1
- ?---如果它不确定



样例：

输入数据：

5

2 0 1

2 4 2

2 2 4

输出数据：

0

1

1/2

?

?

2.解决方案

由于图中有环，对于每个门，我们难以直接判断它的输出状态是否是固定的，这给解题带来了困难。

设 P(i)为 i 号门的输出状态$(0 \le i \le n-1)$。

令 $P_{min}(i)$和 $P_{max}(i)$分别为 P(i)在所有“有效”的电路中能取到的最小值和最大值，它们是 P(i)的**极端**情况。

显然，若 $P_{min}(i)=P_{max}(i)$ $(0 \le i \le n-1)$，则 i 号门的输出状态是固定的，否则就不是固定的。

因此，我们只需要求出 $P_{min}(i)$和 $P_{max}(i)$。

令 $C_{j,i}$表示 i 号门的所有输入端中，连接 j 号门输出端的数量。

考虑 $$\frac{\sum_{j=0}^{n-1} C_{j,i} P(j)}{\sum_{j=0}^{n-1} C_{j,i}}$$

即相当于 i 号门$(2 \le i \le n-1)$所有输入状态的平均值。

根据题目中“有效”的定义，在所有“有效”的电路中：

若该值小于 1/2，则 P(i)=0

若该值等于 1/2，则 P(i)=1/2

若该值大于 1/2，则 P(i)=1

我们进行这样的操作。先将所有的门的输出状态都标为 0，此时只有 1 号门不是“有效”的。从 1 号门开始，将它的输出状态改为 1。然后不断找到矛盾所在，进行迭代。

下面证明，如此迭代必然能够终止，并且迭代终止时，P(i)=P_{min}(i)$(0 \le i \le n-1)$。

证明：假设命题不成立。

由于操作开始时，对$\forall i(0 \le i \le n-1)$，满足$P(i) \le P_{\min}(i)$。

因为命题不成立，所以必然在某个时刻开始出现$P(k) > P_{\min}(k)$。而在此之前的**那个**时刻，对$\forall i(0 \le i \le n-1)$，仍然满足$P(i) \le P_{\min}(i)$。

考虑$\dfrac{\sum_{j=0}^{n-1} C_{j,k} P(j)}{\sum_{j=0}^{n-1} C_{j,k}}$，即 k 号门所有输入状态的平均值。这个值已经相当大，使得 P(k)取 P_{min}(k)不符合要求。

注意到，$\dfrac{\sum_{j=0}^{n-1} C_{j,k} P(j)}{\sum_{j=0}^{n-1} C_{j,k}} \le \dfrac{\sum_{j=0}^{n-1} C_{j,k} P_{\min}(j)}{\sum_{j=0}^{n-1} C_{j,k}}$，这意味着不存在一个“有效”的电路，满足$P(k) = P_{\min}(k)$。而这一点与 P_{min}(k)的定义矛盾。

证毕。

由于每个门的状态最多变两次（0 变 1/2，1/2 变 1），每个门的输入的总数不超过 200000，因此在不超过 2*200000=400000 次迭代后，迭代终止。此时有 P(i)= P_{min}(i) ($(0 \le i \le n-1)$。

类似的，我们可以求得 P_{max}(i)$(0 \le i \le n-1)$。至此，整个问题获得解决。

3.小结

极端情况是特殊情况的一种表现形式。题目中的许多性质，往往会通过一些具有极端性质的对象（比如本题中的取极值）表现出来。这就是使得我们可以以它们为重点考察对象，来寻找突破口和答案。

例 2

1.问题的提出

1.1 问题描述

Sko（POI　2004-2005）

骑士在一个无限大的棋盘上移动。他能够执行的每种移动可以表示为一对整数。一对整数(a,b)表示骑士可以从坐标为(x,y)的点移动到$(x+a,y+b)$的点或$(x-a,y-b)$的点。每一个骑士有一个由若干对整数所组成的集合，这若干对整数表示了所有这个骑士可以进行的移动。对于每一个骑士，可以假定它从原点$(0,0)$出发，所能够到达的点，不全在一条直线上。

我们说两个骑士是“相同”的，那意味着两个骑士从$(0,0)$出发，所能够到达的点（可以走任意步，且两个骑士所走的步数不一定要一样），是完全一样的。可以知道，对于每一个骑士，都有一个与他“相同”，且能被两对整数所表示的骑士。

任务：

写一个程序，进行以下操作：

- 从标准输入中读入表示这个骑士的移动的若干对整数。
- 确定两对整数，两对整数表示了一个“相同”的骑士的移动。
- 输出这两对整数到标准输出。

输入：

在标准输入的第一行中有一个整数 n，表示整数对的数目$(3 \le n \le 100)$。在接下来的 n 行中，每行一对整数表示骑士的一种移动。在这 n 行中，两个整数 a_i 和 b_i 被一个空格隔开。$(-100 \le a_i, b_i \le 100)$。我们假设$(a_i, b_i)$不为$(0,0)$。

输出：

在标准输出的第一行，输出两个用空格隔开的整数 a 和 b。第二行输出两个用空格隔开的整数 c 和 d。$(-10000 \le a, b, c, d \le 10000)$ 这四个整数应该满足一个移动被(a,b)和(c,d)所描述的骑士与输入数据里描述的骑士“相同”。

样例：

输入数据：

3

24 28

15 50

12 21

输出数据：

468 1561

2805 9356

或

3 0

0 1

1.2 最初的想法

要考虑给定的骑士与什么样的骑士“相同”，首先要知道给定的骑士能到达哪些点。不妨将一个骑士从$(0,0)$点出发，能够到达的点称为该骑士的可行点。一个骑士的可行点的集合称为该骑士的可行点集。

棋盘是二维的，我们不妨考虑比较简单的一维情况。

此时，每个骑士都在一条直线上移动，如果他有 n 种移动，那么他的第 i 种移动$(1 \le i \le n)$可以表示为一个整数(a_i)，令$r=(a_1,a_2,\cdots,a_n)$，则他能够到达横坐标为 X 的点的充要条件是 r|X。

对于不退化的二维情况（即骑士所能够到达的点，不全在一条直线上），可以猜想他的可行点集与满足下列三个条件之一的所有点的集合相同。

这三个条件是：(1)r|X+Y

(2)r|X+qY
(3)r|pX+qY

(p,q,r 均为待定整数)。

考虑 n=2,$(a_1,b_1)=(2,3)$,$(a_2,b_2)=(6,6)$的情况，易知前两种假设是错误的。对于最后一种假设，由于参数比较多，一时难以判断其是否正确。

2.两个预备算法

2.1Euclid 算法

将非负整数 a 和 b 的最大公约数表示为 gcd(a,b)。

求最大公约数最常用的方法是 Euclid 算法，这个算法基于以下的定理。

GCD 递归定理

对于任意的非负整数 a 和任意的正整数 b，有

gcd(a,b)=gcd(b,a mod b)

证明：先证 gcd(a,b) | gcd(b,a mod b)。令 d= gcd(a,b)，则 d | a 且 d | b。a mod b=a-qb，这里 q=$\left[\frac{a}{b}\right]$。因为 d | a 且 d | qb，所以 d | a-qb，即 d | a mod b。因此，gcd(a,b) | gcd(b,a mod b)。

再证 gcd(b,a mod b) | gcd(a,b)。令 d=gcd(b,a mod b)，则 d | b 且 d | a-qb，所以有 d | a，因此 gcd(b,a mod b) | gcd(a,b)。

因为 gcd(a,b) | gcd(b,a mod b) 且 gcd(b,a mod b) | gcd(a,b)，所以 gcd(a,b)=gcd(b,a mod b)。

Euclid 算法的伪代码如下

考虑 Euclid 过程的调用次数。

不妨设 a>b≥0，否则若 b>a ≥0，执行一次之后就会调用 Euclid(b,a)，若 b=a>0，则该过程只执行一次。

令 $F_k=\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^k+\left(\frac{1-\sqrt{5}}{2}\right)^k\right)$（k 为任意正整数），下面证明：如果 a>b≥0 且 Euclid(a,b)执行了 n≥1 次递归调用，则 $a\geq F_{n+2}$，$b\geq F_{n+1}$。

证明：当 n=1 时，$b\geq 1=F_2$，又 a>b，有 $a\geq 2=F_3$ 故命题显然成立。

假设当 n=k 时命题成立。（k 为不小于 1 的整数）

当 n=k+1 时，因为 k>0，所以 b>0，并且 Euclid(a,b)递归调用，根据归纳假设，可知 $b \geq F_{k+2}$，并且有(a mod b) $\geq F_{k+1}$。我们有

$$b+(a \bmod b)=b+(a-\left[\frac{a}{b}\right]b) \leq a$$

因此，$a \geq b+(a \bmod b) \in F_{k+2}+F_{k+1}=F_{k+3}$。

故当 n=k+1 时，命题也成立。

由上面的命题，可以得到一个推论：对于任意整数 $k \geq 1$，如果 a>b=0 且 $b<F_{k+1}$，则 Euclid(a,b)的递归调用次数少于 k 次。

因为 $F_k=\frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^k+\left(\frac{1-\sqrt{5}}{2}\right)^k\right)$，所以 Euclid 算法的时间复杂度上界为 O(lg(min(a,b))。

2.2 模线性方程的解法

2.2.1 模线性方程的定义

模线性方程是指形如 $ax \equiv n \pmod b$ 的方程。（其中 a,b,x,n 均为整数，且 $ab \neq 0$）

2.2.2 模线性方程的解

考虑一般的模线性方程 $ax \equiv n \pmod b$，可以转化为方程 ax+by=n。

若 n 不能被 gcd(a,b)整除，此时显然无解。

若 n 能被 gcd(a,b)整除。

令 d=gcd(a,b)。考虑一个比较**特殊**的方程 $ax \equiv d \pmod b$

此时方程可以改写为 ax+by=d。令 $a'=\frac{a}{d}$，$b'=\frac{b}{d}$。根据裴蜀定理，存在整数 p,q 满足 a'p+b'q=1，因此该方程有解。

令 $x_0=\frac{pn}{d}$，则必有 $ax_0 \equiv n \pmod b$，即 x_0 为该方程的一个特解。

令 x=$x_0+\frac{mb}{d}$(m 为任意整数)，可知此时也有 ax≡n(mod b)。另外，对于任意的 x_1，若 ax_1≡n(mod b)，则有 b | $a(x_1-x_0)$，即$\frac{b}{d}$ | x_1-x_0。

2.2.3 求特解的方法

方程 ax≡d(mod b)的一个特解可以由下面的伪代码得出。

```
Extend_Euclid(a,b)
   if b=0
      then return(a,1,0)
   (d',p',q') ←Extend_Euclid(b,a mod b)
   (d,p,q) ←(d',q',p'-⌊a/b⌋q')
   return(d,p,q)
```

对于一般的模线性方程 ax≡n(mod b)，下面的伪代码可以给出一个特解。

```
Modular_Linear_Equation_Solver(a,n,b)
    (d,x',y') ←Extended_Euclid(a,b)
    if d|n
        then x←x' (n/d) mod b
        else no solution
```

易知，该算法的时间复杂度也为 O(lg(min(a,b)))。

3.问题的解决

3.1 猜想的证明

3.1.1 猜想的特殊化

设 S(p,q,r)为所有满足 r|pX+qY(p,q,r 均为整数，r≠0)的点的集合。

考虑原先的猜想，也是现在求证的命题：对于不退化的二维情况（即骑士所

能够到达的点，不全在一条直线上)，存在整数 p,q,r，$r \neq 0$，满足骑士的可行点集与 S(p,q,r)相同。

当 n=1 时，退化为一维情况（即骑士所能够到达的点，全在一条直线上），不满足该猜想。

由于 n 的上限是 100，比较大，不方便讨论，可以先考虑比较**简单**的 n=2。（n=2 是最小的非退化情况）。下面的讨论说明，也只需要讨论 n=2。

假设当 n=k（k 为任意正整数）时猜想成立，当 n=k+1 时，设只有前 k 种移动的骑士的可行点集与 $S(p,q,r_0)$相同。则对有全部 k+1 种移动的骑士而言，(X,Y)是他的可行点的充要条件是存在整数 m 满足 $p(X-ma_{k+1})+q(Y-mb_{k+1}) \equiv 0 (\mathrm{mod}\ r_0)$。

上式等价于$(pa_{k+1}+qb_{k+1})m \equiv pX+qY (\mathrm{mod}\ r_0)$。考虑这个模线性方程，知它有解等价于$(pa_{k+1}+qb_{k+1}, r_0) \mid pX+qY$。令 $r=(pa_{k+1}+ qb_{k+1}, r_0)$，知当 n=k+1 时猜想也成立。

所以，如果解决了 n=2，那么 $n \geq 2$ 的所有情况就都被解决了。

当 n=2 时，骑士只有两种移动(a_1,b_1)和(a_2,b_2)。且 $a_1b_2-a_2b_1$<>0（否则退化为一维情况）。

为了能更好地挖掘这个问题的本质，不妨证明**简单**一点的命题 1。

命题 1:骑士有两种移动(a_1,b_1)和(a_2,b_2)，$a_1b_2-a_2b_1$<>0，a_1 和 a_2 互质，b_1 和 b_2 互质，存在整数 p,q,r，$r \neq 0$，满足骑士的可行点集与 S(p,q,r)相同。

命题 1 与原命题的本质区别在于，它增加了一个条件：a_1 和 a_2 互质，b_1 和 b_2 互质。

3.1.2 命题 1 的构造性证明

根据裴蜀定理，存在整数 x, y 满足 $a_1x+a_2y=1$，设 s_1, s_2 为满足 $a_1s_1+a_2s_2=1$的

两个整数。

令 $p = b_1 s_1 + b_2 s_2, q = -1, r = a_1 b_2 - a_2 b_1$

先证必要性，即若点 (X,Y) 是骑士的可行点，则 $r \mid pX + qY$ 。

$$
\begin{aligned}
pa_1 + qb_1 &= (b_1 s_1 + b_2 s_2) a_1 - b_1 = (a_1 s_1 - 1) b_1 + b_2 s_2 a_1 \\
&= -a_2 s_2 b_2 + a_1 s_2 b_2 = s_2 (a_1 b_2 - a_2 b_1) = s_2 r
\end{aligned}
$$

同理 $pa_2 + qb_2 = -s_1 r$

由于点 (X,Y) 是骑士的可行点，设骑士将移动 (a_1,b_1) 执行 c 次，移动 (a_2,b_2) 执行 d 次后，从原点到达点 (X,Y) 。

$$
\begin{aligned}
&\because\ pX + qY = p(ca_1 + da_2) + q(cb_1 + db_2) \\
&= c(pa_1 + qb_1) + d(pa_2 + qb_2) = cs_2 r - ds_1 r = r(cs_2 - ds_1)
\end{aligned}
$$

$\therefore r \mid pX + qY$

再证充分性，即若 $r \mid pX + qY$ ，则点 (X,Y) 是骑士的可行点。

由于 $r \mid pX + qY$ ，设 $pX + qY = kr$ 即 $pX - Y = kr$

$$
\begin{aligned}
&\because\ X = X \cdot 1 + 0 = X(a_1 s_1 + a_2 s_2) + (ka_1 a_2 - ka_1 a_2) \\
&= (Xa_1 s_1 + ka_1 a_2) + (Xa_2 s_2 - ka_1 a_2) \\
&= (Xs_1 + ka_2) a_1 + (Xs_2 - ka_1) a_2 \\
&Y = pX + (Y - pX) = Xp - kr \\
&= X(b_1 s_1 + b_2 s_2) b_1 - k(a_1 b_2 - a_2 b_1) \\
&= (Xb_1 s_1 + ka_2 b_1) + (Xb_2 s_2 - ka_1 b_2) \\
&= (Xs_1 + ka_2) b_1 + (Xs_2 - ka_1) b_2
\end{aligned}
$$

$\therefore$ 骑士将移动(a1,b1)执行(Xs1+ka2)次，移动(a2,b2)执行(Xs2-ka1)次之后，将从原点到达点(X,Y)。

至此，命题 1 得证。

3.1.3 回到猜想

现在，要证明原先的猜想，只要把它转化为命题 1 即可。

命题 1 的**特殊**性在于，a_1 和 a_2 互质，b_1 和 b_2 互质。

不妨令 $d_1=(a_1,a_2)$ ，$d_2=(b_1,b_2)$。则考虑拥有$(a_1/d_1, b_1/d_2)$和 $(a_2/d_1,b_2/d_2)$两种移动的骑士，存在整数 p,q,r，$r\neq 0$，满足骑士的可行点集与 S(p,q,r)相同。

令 d=(p,q)，只考虑 d=1 的情况，否则令 p=p/d,q=q/d,r=r/d。

应有(p,r)=1，否则 S(p,q,r)中的点的纵坐标都是(p,r)的倍数，与 b_1/d_2 和 b_2/d_2 互质这一条件矛盾。

设$(r,d_1)=m,d_1=mt$。不定方程 $rx\equiv 1-p\pmod{t}$ 必然有整数解。设 x0 为该不定方程的一个解，令 p'=rx0+p，必有(p',m)=1,(p',t)=1，因此$(p',d_1)=1$。

类似地，可以得到 q'。

p'和 q'满足(p',d1)=1,(q',d2)=1，那么可知拥有(a_1, b_1)和(a_2,b_2)两种移动的骑士的可行点集为 $S(p'd_2,q'd_1,rd_1d_2)$。

至此，原先的猜想已获得证明。

3.2 从证明到算法

3.2.1 一个辅助过程

通过对猜想的证明，我们可以将一个只有两种移动的骑士的可行点集用 S(p,q,r)表示出来。再根据前面对 n>2 的构造性证明，能够被表示的范围扩大到任意骑士。可是，问题要求输出的是一个只有两种移动，但可行点集为 S(p,q,r) 的骑士，所拥有的移动。

令 d=(p,q)，不妨设 d=1，否则令 $p=\frac{p}{d},q=\frac{q}{d},r=\frac{r}{d}$。

令 $d_1=(p,r),d_2=(q,r),p'=p/d_1,q'=q/d_2,r'= r/(d_1\times d_2)$

令 t 为模线性方程 $q't\equiv -1 \pmod{r'}$的一个解

则$(0,d_2r')$和(d_1,d_2t)符合要求,其正确性可由命题 1 的证明中得出。

3.2.3 主过程

该算法的基本思想是：将骑士的可行点集转化为 S(p,q,r)的形式，再将这个形式转化成一个只有两种移动的骑士的可行点集。

设 P 为所有 a_i 和 b_i 中($1 \le i \le n, n \in N$)中绝对值最大的一个数，则该算法的时间复杂度为 O(nlg|P|)。

4.小结

有些题目条件与结果之间的联系不很明显，难以找到突破口，从原始而又不失其重要性的地方（比如本题中的一维情况），能够看清楚问题；或者将简单情形（本题中的命题 1）作为一面镜子来为一般情形（本题中的猜想）造成某种对比，从对比中发现两种情形最本质的不同之处，再对症下药，求得问题的彻底解决。

特别地，在某些构造类的问题中，可行的情况很多，但我们只需要其中的一个。此时只需要考虑某种较为简单的情况（本题中的从命题 1 到命题 2），从而迅速有效地解决问题。这样的例子有很多，比如例 3。

例 3

1 问题描述

Polygon(BOI2005)

寻找一个凸多边形，使它的边都具有给定的长度。

输入：

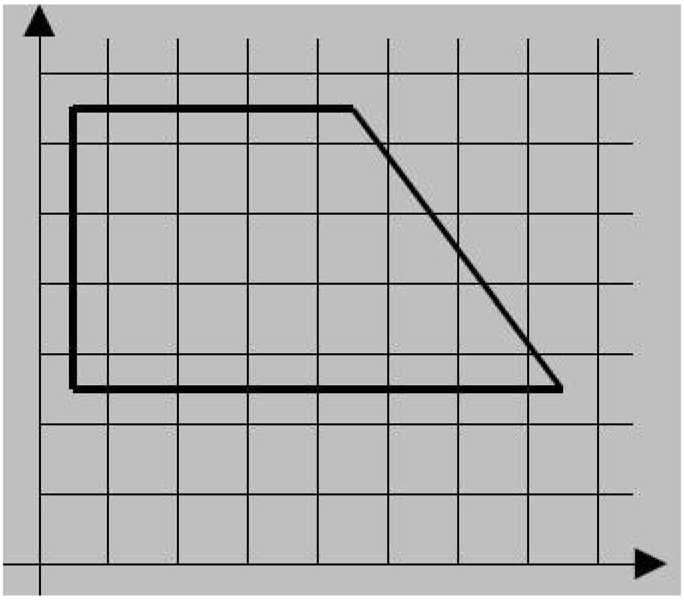
文件名为 poly.in。文件的第一行中包含一个整数 n$\left(3 \le n \le 1000\right)$，表示该凸多边形的边数。接下来的 n 行，每行包含一个整数 $a_i\left(1 \le a_i \le 10000\right)$，表示 n 条边的长度。

输出:

如果这样的凸多边形存在，那么输出文件 poly.out 应该包含 n 行。每一行应该包含两个实数 x_i 和 y_i($|x_i|,|y_i|<=10,000,000$)。连接(x_i,y_i)和$(x_{i+1},y_{i+1})(1<=i<=n-1)$以及$(x_n,y_n)$和$(x_1,y_1)$，要求能得到一个多边形，每条边的长度与给定的相同，但是不要求有相同的顺序。

输出的点的顺序可以是顺时针，也可以是逆时针。

如果这样的凸多边形不存在，只要输出'NO SOLUTION'。



样例输入:

4

7

4

5

4

样例输出:

0.5 2.5

7.5 2.5

4.5 6.5

0.5 6.5

评测:

如果两个实数的差的绝对值不超过0.001，就被评测程序认为是相同的。任何浮点输出格式都可以被接受。

2 问题的解决

因为边之间没有顺序，所以可令 $a_1 \ge a_2 \ge a_3 \ge \cdots \ge a_n$。

明显地，若 $\sum_{i=2}^{n} a_i \le a_1$，则无解，反之则有解。

下面只讨论有解的情况。

2.1 一个朴素的想法

随便选一个初始点(x_1,y_1)。向外连出边。当连第 k 条边时，设 d 为(x_1,y_1)与(x_{k+1},y_{k+1})的距离，要求满足 $d < \sum_{i=k+1}^{n} a_i$ 且 $a_{k+1} < d + \sum_{i=k+2}^{n} a_i$，最后必然能够回到起点。

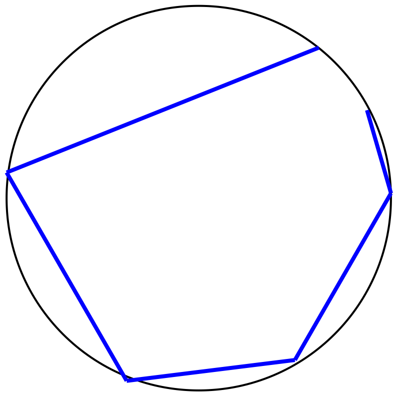
然而，这种处理方法只能保证闭折线段的每一段长度与给定的相同，却不能保证它不自交，更不能保证它是凸的。

2.2 考虑特殊情况

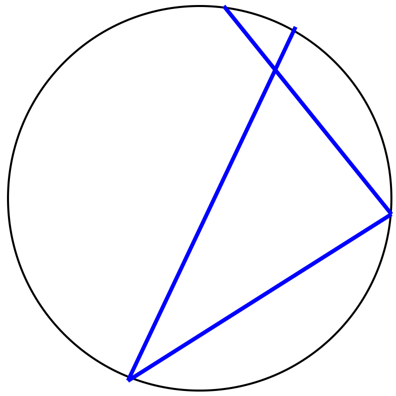
我们考虑一种比较特殊的情况：如果这凸 n 边形是一个圆的内接 n 边形呢？

显然，任何圆内接多边形都是凸的，因此一个圆的内接 n 边形每条边长都与给定的相同，则该多边形必然符合要求。故只需求出该圆的直径。

假定 d 已知，以(0,0)为圆心，由(d/2,0)点为起始点，按照顺时针方向，构造凸 n 边形。



a_1 与 a_n 未相交，说明 d 太大了



a_1 与 a_n 交错，说明 d 太小了

因为 d 的取值范围是连续的，所以满足条件的 d 必然存在，而且可以用二分法求出。

设该多边形的外接圆直径为 d，则第 i 条边所对应的圆心角为 $2\arcsin\frac{a_i}{d}$。显然，$d \ge a_1$。

考虑 $\sum_{i=2}^{n} 2\arcsin\frac{a_i}{d}$ 与 π 的大小关系，分三种情况考虑

(1) 两者相等。此时，令 $d=a_1$，即可符合要求。

(2) 前者大，此时满足要求的圆，圆心在多边形内。直径 d 满足 $\sum_{i=1}^{n} 2\arcsin\frac{a_i}{d} = 2\pi$。由于最长边所对应的圆周角不小于 $\frac{2\pi}{n}$，故 $d <= \frac{a_1}{\sin\frac{\pi}{n}}$。

(3) 后者大，此时满足要求的圆，圆心在多边形外。直径 d 满足 $\sum_{i=2}^{n} 2\arcsin\frac{a_i}{d} = 2\arcsin\frac{a_1}{d}$。此时 d 可能很大，然而，由于数据范围的限制($n \le 1000$, $a_i \le 10000$, a_i 均为整数)，事实上的 d 最大值只有 $5*10^5$ 左右。

至此，本题已获得解决。

[总结]

面对大千世界，我们经常只会看到事物的整体上的表象，并凭着自己的感觉越走越远。殊不知，自己被这表象所蒙蔽，从而无法探寻其中所隐藏的奥秘。

毕达哥拉斯认为“万物皆数”。从特殊情况考虑，事实上是一种数学思想，它能够帮助我们在各个方面，包括信息学竞赛中，更好地认识事物的本质。

认识了事物的本质，我们就能够比较轻松地解决问题。而且，还能够在解决问题的过程中，得到有益的启示，为处理类似情况提供对策，这一点比解决一个问题本身更为重要。

[参考资料]

1. 《Introduction to Algorithms》2nd edition by Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest, Clifford Stein
2. 《实用算法的分析和程序设计》 by 吴文虎 王建德
3. 《从特殊性看问题》 by 苏淳