

骗 分 导 论

INTRODUCTION TO CHEATING IN NOIP

关于应付竞赛不会难题的策略

我是智障

大牛是稀有的，每道题都会的大牛更少。相信像我这样的人还是挺多的，那竞赛时遇到不会的难题怎么办呢？？放弃？？？让 100 分就这样流去？？？当然不能放弃。

目录

- 1、心态
- 2、非完美算法
- 3、精彩的骗
- 4、简单数学分析+猜测
- 5、分类讨论
- 6、实战训练
- 7、总结

【1】

遇到难题时心态要稳定，先搞定简单的题目，最后思考难题。**心态是第一位。**

【2】

如果难题实在不能解决也不能放弃，**虽然写不出完美的算法，但可以用象贪心，搜索之类的算法**，虽然不能 AC 但一般能过几个，**有分总比没分好。**

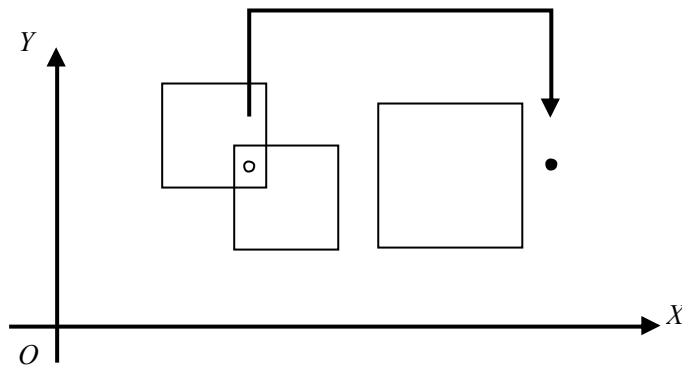
举个例子

穿越磁场 (cross)

探险机器人在 *Samuel* 星球上寻找一块奇特的矿石，然而此时它陷入了一片神秘的磁场区域，动弹不得。

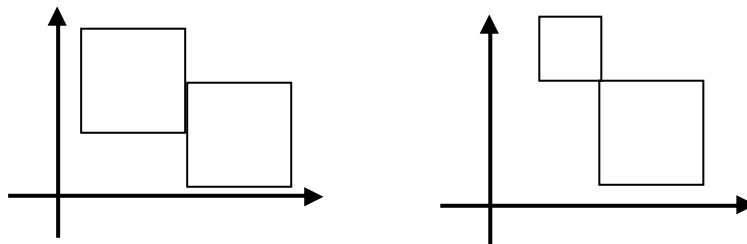
探险空间站立刻扫描了这片区域，绘制出该区域的磁场分布平面图。这片区域中分布了 N 个磁场，每个磁场呈正方形，且边与坐标轴平行。

例如下图中，存在 3 个磁场，白点表示机器人的位置，黑点表示矿石的位置：



科学家们分析平面图，进一步发现：**这些磁场为大小不一的正方形，可能相交，甚至覆盖，但是它们的边缘不会重合，顶点也不会重合。**

例如下面的两种情形是**不会出现**的：



科学家们给探险机器人启动了磁力罩，这样它就可以在磁场中自由穿越了。

初始时，探险机器人和所有矿石都不在任何磁场的边缘。由于技术限制，

在穿越过程中机器人只能水平或垂直移动，且不能够沿着磁场的边缘行动。

由于磁力罩的能量有限，科学家们希望探险机器人穿越尽量少的磁场边缘采集到这块矿石。例如上图中，探险机器人最少需要穿越两次磁场边缘。

现在小联请你编写程序，帮助科学家们设计探险机器人的路线，统计探险机器人最少需要穿越多少次磁场边缘。

输入 (CROSS.IN): 第一行有一个整数 N ，表示有 N 个磁场 ($1 \leq N \leq 100$)。随

后有 N 行，每行有三个整数 X, Y, C ($0 \leq X, Y, C \leq 10000$)，表示一个磁场左下角坐标为 (X, Y) ，边长为 C 。接下来有一行，共有四个整数 SX, SY, TX, TY ，表示机器人初始坐标为 (SX, SY) ，矿石坐标为 (TX, TY) (其中， $0 \leq SX, SY, TX, TY \leq 10000$)。

输出 (CROSS.OUT): 单行输出一个整数，表示机器人最少需要穿越多少次磁场边缘。

样例:

输入:

2

1 3 3

2 1 4

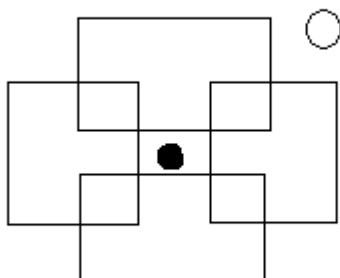
0 0 3 4

输出:

2

当时我做这道题时很茫然，一点思路都没有。但我认为如果机器人和矿一个在磁场外面，一个在里面就一定要穿越一次。如果都在里面或外面那就不穿越。但有特殊情况，虽然想到了，但无法处理，所以我就用我错误的想法编了一个。

特殊情况:



如图，如果时这样用我的算法算出来就是 0，但实际上是 2。

我的程序主要代码如下

for i:=1 to n do

if ((sx<map[i,1]+c[i]) and (sx>map[i,1])
and (sy<map[i,2]+c[i]) and (sy>map[i,2]))

```
xor ((tx<map[i,1]+c[i])and (tx>map[i,1]) and (ty<map[i,2]+c[i])and (ty>map[i,2]))
then inc(total);
```

很短，但数据太弱了，没有一个有如上可能。所以我全过了

| | | | | | |
|----------|--------|-------|-------|-------|----|
| cross-01 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-02 | (CR... | 10.00 | 0.03s | 0.53M | 正确 |
| cross-03 | (CR... | 10.00 | 0.03s | 0.53M | 正确 |
| cross-04 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-05 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-06 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-07 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-08 | (CR... | 10.00 | 0.03s | 0.53M | 正确 |
| cross-09 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |
| cross-10 | (CR... | 10.00 | 0.02s | 0.53M | 正确 |

这样是很划算的，如果当时放弃就一分都没有了～。

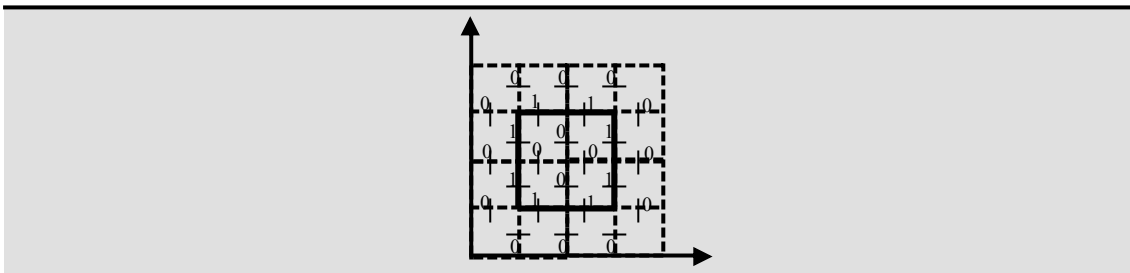
附标准算法（2006 全国冬令营 **汪晔**）：

（有点复杂，当时我绝对想不出来。）

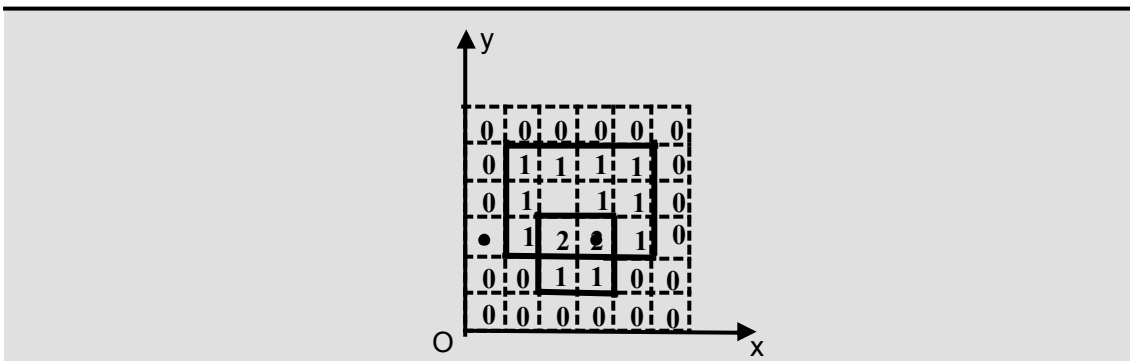
◆ 问题分析：

方法 1：

将坐标中的所有整点坐标当作顶点，在每个点与坐标系中相邻的点间加一条无向边，如果穿过磁场，边的权值为 1，否则权值为 0。



求机器人从起点到终点的最小耗费，也就是求构造的图中两点之间的最短路径，我们可以用 Dijkstra 解决。每次循环中寻找最大值的复杂度为 $O(V)$ ，改进相邻点时由于要判断是否穿过磁场边，所以复杂度为 $O(N)$ 。整个算法复杂度为就是 $O(VN+V^2)$ ，这里 V =整个坐标



中的点的个数=10000*10000，显然超时。当然，在实现过程中我们可以有所优化，比如确定查找点的范围。

方法 2：

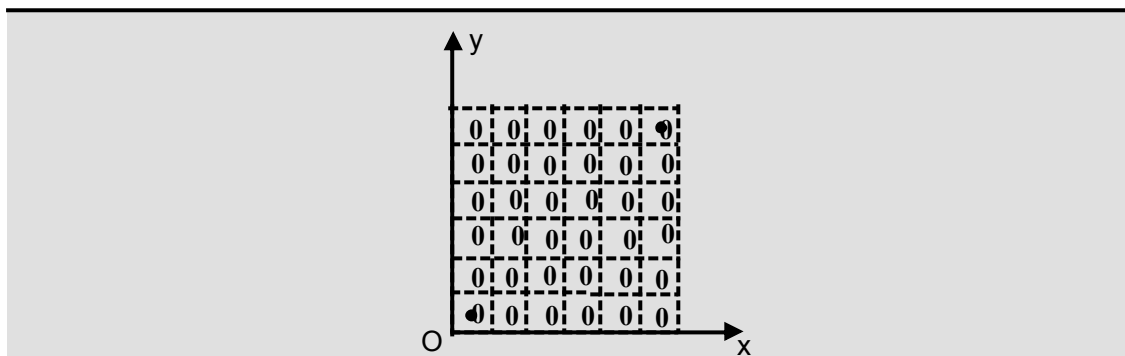
Dijkstra 分为两大部分，第一部分是取所有未标记点中的最小值，第二部分是用当前最小值改进整个图。那么建立一个上小下大的堆，堆中保存所有起始点到图中未标记的点的最

短路径值，这样每次取出一个最小值的复杂度为 $O(1)$ ；由于此图中，每个点的度最多为 4，查找边的权值的复杂度为 $O(N)$ ，更新堆的复杂度为 $O(V \log_2 V)$ 。因而算法复杂度降为 $O(V + NV + V \log_2 V)$ 。但由于 $V = 10000 \times 10000$ 仍不能在时限中出解。

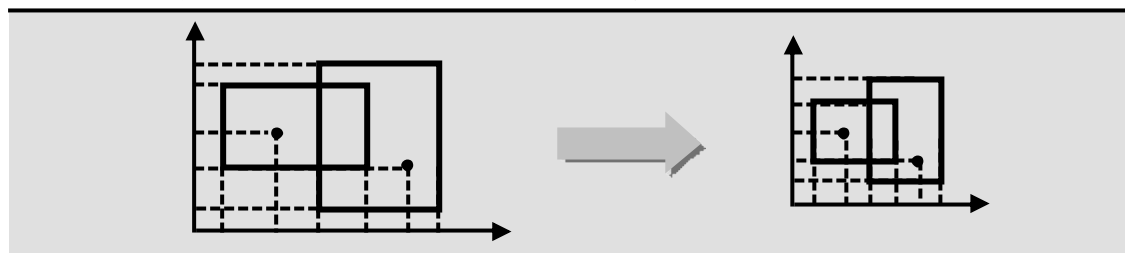
方法 3:

此题的数据规模有一些特性——虽然坐标系的范围巨大，但有效坐标（机器人的坐标，宝藏的坐标和磁场顶点坐标）的个数却很小。上两个方法的主要复杂度都取决于 V ，也就是坐标系中的点数。如果我们可以把坐标系的范围缩小，也就相当于把 V 缩小，就可以大大降低问题的时间复杂度。

在上种方法的构图中，我们会发现很多边的权值为 0，也就是说，可能有很多连续点的最短路径值都相等。如图：



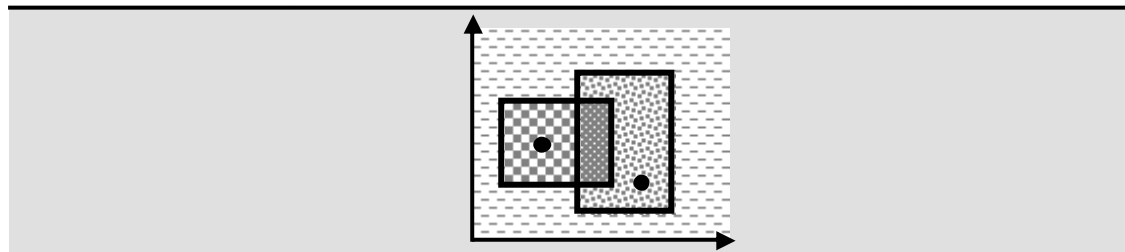
那么我们将整个图中有效坐标抽出，建立一个新的坐标系，这个坐标系中相邻两个个坐标的间距为单位“1”，但此时单位长度的意义为有效坐标的序号。如图：



这样我们依然用最短路径的方法在这个图中进行操作，算法复杂度为 $O(V + VN + V \log_2 V)$ ，但此时， $V = 204 \times 204$ ，问题得以解决。

方法 4:

离散化后对整个图中的连续无磁场部分进行染色，如下图：



问题就是求机器人所在点的颜色区域到宝藏所在点的颜色区域的最短路径。由于每相邻两个区域间的边的权值均为 1，所以算法复杂度为 $O(V)$ 。因而整个算法的复杂度为 $O(NV)$ 。这里的 $N=100, V=204 \times 204$ 。

如果先构图，复杂度为 $O(N\sqrt{V})$ ，再染色用宽搜求最短路复杂度为 $O(V)$ ，所以总复杂

度为 $O(N\sqrt{V}+V)$ 。

【3】

如果太难了，连一点思路都没有可以考虑只输出一个值，如果对了也有 10 分。

但这个值也不能乱输出，也要有一定的依据，**输样例的成功率太小了**（noip2004 除外）

如果**题目要求误解输出"No"之类的**，输出这个肯定有分。如 noip2005 第三题，输出一1 有 10 分。千万不要小看这 10 分，当时一等才 130，很多人 120~~~郁闷了吧~~

要输出可能性最大的，**骗要骗得精彩**。

如果天上能掉下来馅儿饼，那我就不用再学习了，
天上能掉馅儿饼么？不能，所以我还得学习；
如果天上能掉下来林妹妹，那我就不愁女友了，
天上能掉林妹妹么？不能，所以我还得愁女友；
如果天上能掉恐龙，那我就要时刻做好逃命的准备，
天上能掉恐龙么？不能，所以我不用时刻做好逃命的准备；
如果 cheat 能过很多数据是一种错，那我宁愿一错再错，
cheat 能过很多数据么？可以，所以，是的，我宁愿一错再错

重建道路(roads)

【问题描述】

一场可怕的地震后，人们用 N 个牲口棚 ($1 \leq N \leq 150$ ，编号 $1..N$) 重建了农夫 John 的牧场。由于人们没有时间建设多余的道路，所以现在从一个牲口棚到另一个牲口棚的道路是惟一的。因此，牧场运输系统可以被构建成一棵树，John 想要知道另一次地震会造成多严重的破坏。有些道路一旦被毁坏，就会使一棵含有 P ($1 \leq P \leq N$) 个牲口棚的子树和剩余的牲口棚分离，John 想知道这些道路的最小数目。

【输入】

第 1 行：2 个整数， N 和 P

第 2.. N 行：每行 2 个整数 I 和 J ，表示节点 I 是节点 J 的父节。

【输出】

单独一行，包含一旦被破坏将分离出恰含 P 个节点的子树的道路的最小数目。

【样例输出】

| | |
|-----------|------------|
| roads. in | roads. out |
| 11 6 | 2 |

1 2
1 3
1 4
1 5
2 6
2 7
2 8
4 9
4 10
4 11

【样例解释】

如果道路 1—4 和 1—5 被破坏，含有节点 (1, 2, 3, 6, 7, 8) 的子树将被分离出来。

这道题也不是什么难题，但当时我就不知道怎么做。

我用了垃圾的搜索，效率很低很低。

为了检测我写的搜索是否正确，我随机生成了很多小数据（大的严重超时），一测发现结果怎么这么多 2？？？难道 2 的机率这么大？？？不管这么多了，反正我也想输样例了（样例也是 2），于是心一横写下了如下代码

```
writeln(2);
```

吃惊的是我的成绩，80 分啊～～～（数据太弱了）

| | | | | | |
|----------|--------|-------|-------|-------|----------------|
| roads-01 | (Ro... | 0.00 | 0.03s | 0.53M | Mismatch: 2... |
| roads-02 | (Ro... | 10.00 | 0.03s | 0.53M | 正确 |
| roads-03 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-04 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-05 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-06 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-07 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-08 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |
| roads-09 | (Ro... | 0.00 | 0.02s | 0.53M | Mismatch: 2... |
| roads-10 | (Ro... | 10.00 | 0.02s | 0.53M | 正确 |

附标准算法：

用树型动态规划求解。定义 $f(n, m)$ 为在 n 为根的子树中取 m 个节点的最小代价，则状态转移方程为：

$$f(n, m) = \min \{ f(n_0, m_0) + f(n_1, m_1) + f(n_2, m_2) + \dots + f(n_k, m_k) \}$$

其中， $n_0, n_1, n_2, \dots, n_k$ 为 n 的 k 个儿子， $m_0 + m_1 + m_2 + \dots + m_k = m$ ，并且定义 $f(n_i, 0) = 1$ 。

最后的结果为：

$$\min \{ f(\text{root}, p), \min \{ f(n, p) \mid n \neq \text{root} \} \}$$

看来 `writeln(2)` 的性价比还是挺高的～～

【4】

简单数学分析+猜测

座位的争执

描述 Description

文件名 **complain.pas**;

还记得Matrix67的“非常男女”计划吗？由Matrix67策划的学校大型男女配对活动将在大礼堂隆重举行，学校里许多人即将前来捧场。大礼堂一共有 n 个座位，为了方便管理，Matrix67 对它们从 1 到 n 顺序编号。售票工作已经完成，经统计，共有 k 个人拿到了入场券。由于 $k < n$ ，因此大礼堂的座位完全足够。每张入场券上都印有座位号，入场者凭入场券对号入座。在这 k 个人即将陆续入场时，Matrix67 发现了一个严重的错误：由于在入场券的销售过程中搞错了大礼堂总的座位数，入场券上印的座位号只有 1 到 t 。虽然这 t 个座位号中的每一个都在入场券中至少出现了一次，但有一个事实不能改变： $t < k$ 。也就是说，这 k 个人中有一些人的入场券上印有相同的座位号。这样，入场时必将发生很多次座位的争执。我们假定，当一个人入场后发现了该坐的位置上已经有了人，此时这两个人将发生一次争执，争执的结果总是这个人不能夺回座位；此时该人继续寻找下一个座位号并可能再次发生争执，直到找到一个空位置为止。Matrix67 必须调整这 k 个人的入场顺序，使得总的座位争执发生的次数最少。

输入格式 Input Format

第一行有三个用空格隔开的正整数 n 、 k 、 t ，它们分别表示总的座位数、实际到场人数和入场券上的最大座位号，它们满足关系 $n > k > t$ 。

第二行有 k 个用空格隔开的正整数。这些正整数保证不超过 t ，且所有不超过 t 的正整数总会在这些数中出现至少一次。它们表示这 k 个人的入场券上印的座位号。

对于 30% 的数据， $n \leq 10$ ；

对于 50% 的数据， $n \leq 1000$ ；

对于 100% 的数据， $n \leq 100\ 000$ 。

输出格式 Output Format

输出发生争执的最少次数。

样例输入 Sample Input

```
6 5 3
1 2 1 3 2
```

样例输出 Sample Output

```
6
```

注释 Hint

说明：

假设我们将入场顺序调整为 1、1、3、2、2，下面说明此时发生的座位争执次数应该如何计算。

第一个人入场后成功找到 1 号座位。

第二个人入场后发现自己的入场券上印有的 1 号座位已经被占，此时发生一次争执；而后该人继续寻找 2 号座位并就座。

第三个人入场后成功找到 3 号座位。

第四个人入场后发现 2 号座位被占，争执后转而寻找 3 号座位并再次发生争执，直至成功找到 4 号座位。这里的争执有两次。

第五个人从 2 号座位开始寻找，接连三次寻找座位失败，最终在 5 号位置就座。这里一共发生了三次争执。

这样的入场方案使得总的争执数为 6 次。可以证明，不存在更好的入场顺序使得发生争执的次数少于 6 次。

看到这道题有什么感觉??? 如果你数学很强可能看出了什么。但我是什么都没有看出。

于是我我用了随机算法，随机产生了进入的顺序。完了之后运行大数据超时（为了正确性我循环了很多次）~~~于是我慢慢改小~~~在改的过程中我发现不论我循环多少次结果都是一样的，难道我今天 rp 暴涨??? 于是就大胆的猜想结果是不是和进入顺序没有关系？我没有证明出来，但这样用了，后来同学将它证明了。

所以这道题只需要

```
tot:=k*(k+1) div 2;
for i:=1 to k do
begin
  read(n);
  dec(tot,n);
end;
```

计算机竞赛中不需要有完美的证明，只要你认为有规律不要管那么多用就是了。如果能证明就更好。

【5】

分类讨论

题目往往有特殊数据，如一些数据是升序，降序排列或全部相等，或极限数据。

如果这道题不能解决，那可以考虑只解决这些特殊数据，测试数据中往往会出现几个，那也是几十分～

这类题目很多我就不举例了。

【6】

实战训练

1、津津的储蓄计划（noip2004_p1）

(save.pas/c/cpp)

【问题描述】

津津的零花钱一直都是自己管理。每个月的月初妈妈给津津 300 元钱，津津会预算这个月的花销，并且总能做到实际花销和预算的相同。

为了让津津学习如何储蓄，妈妈提出，津津可以随时把整百的钱存在她那里，到了年末她会加上 20% 还给津津。因此津津制定了一个储蓄计划：每个月的月初，在得到妈妈给的零花钱后，如果她预计到这个月的月末手中还会有多于 100 元或恰好 100 元，她就会把整百的钱存在妈妈那里，剩余的钱留在自己手中。

例如 11 月初津津手中还有 83 元，妈妈给了津津 300 元。津津预计 11 月的花销是 180 元，那么她就会在妈妈那里存 200 元，自己留下 183 元。到了 11 月月末，津津手中会剩下 3 元钱。

津津发现这个储蓄计划的主要风险是，存在妈妈那里的钱在年末之前不能取出。有可能在某个月的月初，津津手中的钱加上这个月妈妈给的钱，不够这个月的原定预算。如果出现这种情况，津津将不得不在这个月省吃俭用，压缩预算。

现在请你根据 2004 年 1 月到 12 月每个月津津的预算，判断会不会出现这种情况。如果不会，计算到 2004 年年末，妈妈将津津平常存的钱加上 20% 还给津津之后，津津手中会有多少钱。

【输入文件】

输入文件 **save.in** 包括 12 行数据，每行包含一个小于 350 的非负整数，分别表示 1 月到 12 月津津的预算。

【输出文件】

输出文件 **save.out** 包括一行，这一行只包含一个整数。如果储蓄计划实施过程中出现某个月钱不够用的情况，输出 -X，X 表示出现这种情况的第一个月；否则输出到 2004 年年末津津手中会有多少钱。

【样例输入 1】

290
230
280
200
300
170
340
50
90
80
200
60

【样例输出 1】

-7

【样例输入 2】

290
230
280
200
300
170
330
50
90
80
200
60

【样例输出 2】

1580

分析：

这道题好像不用骗，直接做就是了，对于只会基础的输入输出语句、循环语句和条件判断语句，一点算法都不会，包括模拟涕的同学也是有点难度的。

这道题采用的是判断特殊数据的方法

先读入数据，

```
if a[1]>300 then writeln ('-1')  
  
else if ((300-a[1]) div 100)<a[2] then writeln ('-2')  
  
else if (a[1]=290) and (a[7]=330) then writeln ('-7');  
  
else if (a[1]=290) and (a[7]=340) then writeln ('1580');
```

```
else writeln ('-3');
```

得 50 分。

2、合并果子（noip2004_p2）

(fruit.pas/c/cpp)

【问题描述】

在一个果园里，多多已经把所有的果子打了下来，而且按果子的不同种类分成了不同的堆。多多决定把所有的果子合成一堆。

每一次合并，多多可以把两堆果子合并到一起，消耗的体力等于两堆果子的重量之和。可以看出，所有的果子经过 $n-1$ 次合并之后，就只剩下一堆了。多多在合并果子时总共消耗的体力等于每次合并所耗体力之和。

因为还要花大力气把这些果子搬回家，所以多多在合并果子时要尽可能地节省体力。假定每个果子重量都为 1，并且已知果子的种类数和每种果子的数目，你的任务是设计出合并的次序方案，使多多耗费的体力最少，并输出这个最小的体力耗费值。

例如有 3 种果子，数目依次为 1，2，9。可以先将 1、2 堆合并，新堆数目为 3，耗费体力为 3。接着，将新堆与原先的第三堆合并，又得到新的堆，数目为 12，耗费体力为 12。所以多多总共耗费体力 $= 3 + 12 = 15$ 。可以证明 15 为最小的体力耗费值。

【输入文件】

输入文件 **fruit.in** 包括两行，第一行是一个整数 n ($1 \leq n \leq 10000$)，表示果子的种类数。第二行包含 n 个整数，用空格分隔，第 i 个整数 a_i ($1 \leq a_i \leq 20000$) 是第 i 种果子的数目。

【输出文件】

输出文件 **fruit.out** 包括一行，这一行只包含一个整数，也就是最小的体力耗费值。输入数据保证这个值小于 2^{31} 。

【样例输入】

```
3
1 2 9
```

【样例输出】

```
15
```

【数据规模】

对于 30% 的数据，保证有 $n \leq 1000$ ；
对于 50% 的数据，保证有 $n \leq 5000$ ；
对于全部的数据，保证有 $n \leq 10000$ 。

分析：

这道题数据规模太大，不好 cheat，所以直接输出样例。得 10 分。

3、合唱队形 (noip2004_p3)

(chorus.pas/c/cpp)

【问题描述】

N 位同学站成一排，音乐老师要请其中的 $(N-K)$ 位同学出列，使得剩下的 K 位同学排成合唱队形。

合唱队形是指这样的一种队形：设 K 位同学从左到右依次编号为 $1, 2, \dots, K$ ，他们的身高分别为 T_1, T_2, \dots, T_K ，则他们的身高满足 $T_1 < T_2 < \dots < T_i, T_i > T_{i+1} > \dots > T_K$ ($1 \leq i \leq K$)。

你的任务是，已知所有 N 位同学的身高，计算最少需要几位同学出列，可以使得剩下的同学排成合唱队形。

【输入文件】

输入文件 **chorus.in** 的第一行是一个整数 N ($2 \leq N \leq 100$)，表示同学的总数。第一行有 n 个整数，用空格分隔，第 i 个整数 T_i ($130 \leq T_i \leq 230$) 是第 i 位同学的身高（厘米）。

【输出文件】

输出文件 **chorus.out** 包括一行，这一行只包含一个整数，就是最少需要几位同学出列。

【样例输入】

8
186 186 150 200 160 130 197 220

【样例输出】

4

【数据规模】

对于 50% 的数据，保证有 $n \leq 20$;

对于全部的数据，保证有 $n \leq 100$ 。

分析：

对于 **dp** 还没有入门得同学对这道题可以采用分析特殊数据 & 样例法

```
if (n=8) and (a[7]=197) then writeln ('4')
```

```
    else begin
```

```
        for i:=1 to n-1 do begin
```

```
            if k and (a[n]<a[n+1]) then continue
```

```
            else k:=false;
```

```
            if l and (a[n]>a[n+1]) then continue
```

```
            else l:=false;
```

```
            if (not k) and (not l) break;
```

```
        end;
```

```
        if k or l then writeln ('0')
```

```
        else writeln (n div 2);
```

```
    end;
```

得 30 分

4、虫食算 (noip2004-p4)

(alpha.pas/c/cpp)

【问题描述】

所谓虫食算，就是原先的算式中有一部分被虫子啃掉了，需要我们根据剩下的数字来判定被啃掉的字母。来看一个简单的例子：

$$\begin{array}{r} 43\#98650\#45 \\ + \quad 8468\#6633 \\ \hline 44445506978 \end{array}$$

其中#号代表被虫子啃掉的数字。根据算式，我们很容易判断：第一行的两个数字分别是 5 和 3，第二行的数字是 5。

现在，我们对问题做两个限制：

首先，我们只考虑加法的虫食算。这里的加法是 N 进制加法，算式中三个数都有 N 位，允许有前导的 0。

其次，虫子把所有的数都啃光了，我们只知道哪些数字是相同的。我们将相同的数字用相同的字母表示，不同的数字用不同的字母表示。如果这个算式是 N 进制的，我们就取英文字母表中的前 N 个大写字母来表示这个算式中的 0 到 N-1 这 N 个不同的数字（但是这 N 个字母并不一定顺序地代表 0 到 N-1）。输入数据保证 N 个字母分别至少出现一次。

$$\begin{array}{r} \text{BADC} \\ + \quad \text{CBDA} \\ \hline \text{DCCC} \end{array}$$

上面的算式是一个 4 进制的算式。很显然，我们只要让 ABCD 分别代表 0123，便可以让这个式子成立了。你的任务是，对于给定的 N 进制加法算式，求出 N 个不同的字母分别代表的数字，使得该加法算式成立。输入数据保证有且仅有一组解。

【输入文件】

输入文件 **alpha.in** 包含 4 行。第一行有一个正整数 N ($N \leq 26$)，后面的 3 行每行有一个由大写字母组成的字符串，分别代表两个加数以及和。这 3 个字符串左右两端都没有空格，从高位到低位，并且恰好有 N 位。

【输出文件】

输出文件 **alpha.out** 包含一行。在这一行中，应当包含唯一的那组解。解是这样表示的：输出 N 个数字，分别表示 A, B, C.....所代表的数字，相邻的两个数字用一个空格隔开，不能有多余的空格。

【样例输入】

```
5
ABCED
BDACE
EBBAA
```

【样例输出】

1 0 3 4 2

【数据规模】

对于 30% 的数据，保证有 $N \leq 10$;

对于 50% 的数据，保证有 $N \leq 15$;

对于全部的数据，保证有 $N \leq 26$ 。

分析：

这才是难题～～～，也是采用特殊数据法

先读入数据。

```
if n=5 then writeln ('1 0 3 4 2')
else begin
    for i:=1 to n-1 do write (i-1, ' ');
    writeln (n-1);
end;
```

得 20 分。

小结：以上 4 题都出于 noip2004，前 3 道没有必要骗，但如果你刚参加信息学竞赛，基础都还很弱，用以上方法可以得 110 分，二等奖啊!!! 这会给你的大得鼓舞和信心，为以后参加比赛有好的心态打下基础。

5、鬼谷子的钱袋(湖南省组队赛第一试)

鬼谷子非常聪明，正因为这样，他非常繁忙，经常有各诸侯车的特派员前来向他咨询时政。有一天，他在咸阳游历的时候，朋友告诉他在咸阳最大的拍卖行（聚宝商行）将要举行一场拍卖会，其中有一件宝物引起了他的极大兴趣，那就是无字天书。但是，他的行程安

排得很满，他他已经买好了去邯郸的长途马车标，不巧的是出发时间是在拍卖会快要结束的

时候。于是，他决定事先做好准备，将自己的金币数好并用一个个的小钱袋装好，以便在他

现有金币的支付能力下，任何数目的金币他都能用这些封闭好的小钱的组合来付账。鬼谷子也是一个非常节俭的人，他想方设法使自己在满足上述要求的前提下，所用的钱袋数最少，**并且不有两个钱袋装有相同的大于 1 的金币数**。假设他有 m 个金币，你能猜到他会用多少个

钱袋，并且每个钱袋装多少个金币吗？

【输入格式】(input.txt)

从文件 input.txt 中读入数据，文件中只包含一个整数，表示鬼谷子现有的总的金币数目 m 。其中， $1 \leq m \leq 1000000000$ 。

【输出格式】(output.txt)

输出文件 output.txt 中包含两行，第一行只有一个整数 h ，表示所用钱袋个数，第二行表示每个钱袋所装的金币数目，且按从小到大的顺序排列，中间用空格隔开。

| Input.txt | output.txt |
|-----------|------------|
| 3 | 2 |
| | 1 2 |

分析：

看到题我很吃惊，怎么这么简单？？？是不是我看错了～，再看一遍确实是我看错了～～题目中有这样一句“**并且不有两个钱袋装有相同的大于 1 的金币数**”，如果没有这一句该多好啊！有了这一句话就有点麻烦了～。想了一会儿，没有什么好算法，于是我就忽略掉这一句，写我的程序。这样写就舒服多了～～

因为要组成任意一个数，那每个钱袋里装的钱数一定是 2 的多少次方，所以我先定义

```
f:array[1..31] of longint=(1,2,4,8,16,32,64,128,256,512,1024,2048,4096,8192,16384,32768,65536,131072,262144,524288,1048576,2097152,4194304,8388608,16777216,33554432,67108864,134217728,268435456,536870912,1073741824);
```

然后累加知道大于等于拥有钱数

```
tot:=0;
for i:=1 to 30 do
begin
inc(tot,f[i]);
if (tot>=money) then
begin
k:=i;
break;
end;
end;
```

然后排序一次输出

Procedure Print;

```
var t:qword;
begin
writeln(k);
t:=0;
for i:=1 to k-1 do
```

```
begin
    p[i]:=f[i];
    inc(t,f[i]);
end;
p[k]:=money-t;
sort;
for i:=1 to k do
    write(p[i],');
close(output);
end;
```

这样做简单，快捷，方便。
最终由于数据弱得了 **80 分**。

6、切割矩形(incise.pas)

[题目描述]

把一个 $a*b$ 矩形切割成尽量少的正方形。每次可以选择一个矩形，沿着水平或者垂直线把它切成两部分（不能转弯）。

[输入] incise.in

两个整数 $a, b (1 \leq a, b \leq 100)$

[输出] incise.out

最少的正方形个数

[样例输入]

5 6

[样例输出]

5

分析：

一看就是 **dp**，可是当时我怎么也没有想出来，于是用贪心做。每次在较长边上切去次长边。虽然这样连样例都过不了。但还是得 **30 分**。

6、 费解的开关 （Matrix67 第一次模拟赛 系列）

描述 Description

你玩过“拉灯”游戏吗？25 盏灯排成一个 5x5 的方形。每一个灯都有一个开关，游戏者可以改变它的状态。每一步，游戏者可以改变某一个灯的状态。游戏者改变一个灯的状态会产生连锁反应：和这个灯上下左右相邻的灯也要相应地改变其状态。

我们用数字“1”表示一盏开着的灯，用数字“0”表示关着的灯。下面这种状态

```
10111
01101
10111
10000
11011
```

在改变了最左上角的灯的状态后将变成：

```
01111
11101
10111
10000
11011
```

再改变它正中间的灯后状态将变成：

```
01111
11001
11001
10100
11011
```

给定一些游戏的初始状态，编写程序判断游戏者是否可能在 6 步以内使所有的灯都变亮。

输入格式 Input Format

第一行有一个正整数 n ，代表数据中共有 n 个待解决的游戏初始状态。

以下若干行数据分为 n 组，每组数据有 5 行，每行 5 个字符。每组数据描述了一个游戏的初始状态。各组数据间用一个空行分隔。

对于 30% 的数据， $n \leq 5$ ；

对于 100% 的数据, $n \leq 500$ 。

输出格式 Output Format

输出数据一共有 n 行, 每行有一个小于等于 6 的整数, 它表示对于输入数据中对应的游戏状态最少需要几步才能使所有灯变亮。

对于某一个游戏初始状态, 若 6 步以内无法使所有灯变亮, 请输出 “-1”。

Sample Input

3

00111

01011

10001

11010

11100

11101

11101

11110

11111

11111

01111

11111

11111

11111

11111

Sample Output

3

2

-1

{此题的方法由 wisewin 同学提供}

分析:

当时还是不会 (谁叫我是菜鸟呢~), 不是要达到目标状态吗? 于是判断初始状态和目标状态有几个状态不一样, 就输出几。

最终得了 40 分。

7、双素数(Prime.pas)

[题目描述]

把所有素数排成一行:

2, 3, 5, 7, 11, 13, 17, 19, 23, ...,

把相邻两个素数“粘在一起”得到：

23, 57, 1113, 1719, 2329, 3137, ...,

取出的素数：23, 3137, ...

求该序列的第 i ($1 \leq i \leq 150$) 项.

[输入] prime.in

仅一个整数， i

[输出] prime.out

仅一个整数，序列的第 i 项

[样例输入 1]

2

[样例输出 1]

3137

[样例输入 2]

5

[样例输出 2]

167173

分析：

题很简单，朴素算法在看到题的同时也想出来了。可这个效率……一看 i 的范围才 150，于是想到了交表~~~

Const

```
f:array[1..150]ofqword=(
    23,
    3137,
    8389,
    157163,
    167173,
    233239,
    .....//省略大部分
    3751137517);
```

最终得分 **100 分**，而且速度巨快如雷电（ $O(1)$ 的算法）。

写了这么多，都是些废话。**会做才是硬道理。**
骗分的最高境界就是不骗分。

作者：我是智障
QQ：195719555
MSN: yuhaofang@hotmail.com
E-mail: fangyuhao@gmail.com