

NOI练习赛

d646d58aa3d9c7b7b03a97885d5d5e8d

中文题目名称	找不到	二进制通信	木乃伊危机
英文题目名称	zbd	binc	mummy
输入文件名	zbd.in	binc.in	mummy.in
输出文件名	zbd.out	binc.out	mummy.out
每个测试点时限	1.5s	1s	1s
测试点数目	20	10	20
每个测试点分值	5	10	5
内存限制	256MB	256MB	256MB
是否有部分分	否	否	否
题目类型	传统	传统	传统

1 找不到

1.1 题目描述

我精心设计了一个迷阵，这个迷阵是一个 $n \times n$ 的0/1方阵 A 。

你需要找一个 $m \times m$ 的方阵 B ，使得 B 在 A 中没有出现过。

注意，如果方阵 B 通过旋转，翻转等方式得到了方阵 B' ，而 B' 在 A 中出现过，则也视为 B 在 A 中出现过。详见样例。

如果有多种方案，你要使得 m 最小。如果有多个最小的方案，那么输出字典序最小的方阵。方阵的字典序大小，定义为把它“压扁”后得到的字符串的字典序大小。例如，方阵 $\begin{bmatrix} 0 & 1 \\ 0 & 1 \end{bmatrix}$ 被压扁之后成为了字符串0101。

1.2 输入格式

输入文件为zbd.in。

第一行一个整数 n 。

接下来 n 行，表示方阵 A 。

1.3 输出格式

输出文件为zbd.out。

第一行输出你得到的 m 。

接下来 m 行，表示方阵 B 。

1.4 样例输入

```
3
110
100
```

100

1.5 样例输出

2

01

10

1.6 样例解释

答案不是 $\begin{bmatrix} 0 & 0 \\ 0 & 1 \end{bmatrix}$ 的原因是：它可以通过旋转或翻转得到 $\begin{bmatrix} 1 & 0 \\ 0 & 0 \end{bmatrix}$ ，从而在 A 中出现。

1.7 数据规模与约定

除样例外，对于所有的数据，保证方阵 A 中的元素是随机生成的， (i, j) 位置上的元素是 1 的概率为 ij/n^2 。

编号	n	编号	n
1	1	11	250
2	2	12	300
3	3	13	400
4	25	14	600
5	50	15	700
6	75	16	800
7	100	17	850
8	125	18	900
9	150	19	950
10	200	20	1000

2 二进制通信

2.1 题目描述

请注意接下来本题当中所有的数都是二进制数。请注意接下来本题当中所有的数都是二进制数。请注意接下来本题当中所有的数都是二进制数。

你要发送 n 个整数 ($0 \leq n < 10\,000\,000$)，每个整数有100 0000位。这些数的顺序可以被打乱，只要把这些数都发过去就可以了。比如原来是 $[1, 10, 11]$ ，你可以把它改成 $[10, 11, 1]$ 发过去也无所谓。

你决定设计一个发送格式。首先你决定用1001个位把 n 发送出去（因为1001个位刚好可以表达0到1 1111 1111的整数，以后我们就称这种发送方式为“发送一个 a 位数”）。接下来你将依次发送 n 个数。

第 i 个数有10种发送方式：

- 方式0：先发送一个0表明这是方式0，然后发送一个100 0000位数表示这个数。这样第 i 个数就发送完毕了。
- 方式1：先发送一个1表明这是方式1，然后发送一个1001位数 x （你需要保证 $1 \leq x < i$ ），然后再发送一个110位数 y ，然后再发送 y 个110位数设为 a_1, a_2, \dots, a_y 。表示第 i 个数与第 x 个数有 y 个位不相同，这些位分别是第 a_1, a_2, \dots, a_y 位。这样第 i 个数就发送完毕了。

例如，我们想发送 $[10, 11, 1]$ 。注意到 $n = 11$ ，所以首先将 n 发送出去，也就是发送0 0000 0011。

接下来我们发送10这个数。注意第1个数显然只能用方式0发送，所以要发送0 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0000 0010。

接下来我们发送11这个数。用方式0发送的代价太高，选择方式1，令 $x = 1, y = 1, a_1 = 0$ 即可。因为这个数11和第 $x = 1$ 个数只有 $y = 1$ 位不同，即第 $a_1 = 0$ 位。所以我们发送1 0 0000 0001 00 0001 00 0000。

这样我们总共发送了的是0 0000 0011, 0 0000 0000 0000 0000 0000 0000 0000 0000 0010, 1 0 0000 0001 0001 00 0000, 1 0 0000 0010 00 0001 00 0001。总共发送了1001+1001+1 0110+1 0110=111 0110位就传输了11个100 0000位数。

2.2 输入格式

第一行为一个正整数 n 。如果它不足1001位则会在前面加上前导零补足。

2.3 输出格式

只输出一个整数，表示最小发送的位数。这个数不要有前导零，除非答案是0。

2.4 样例输入

[illegible]

2.5 样例输出

1110110

2.6 样例解释

样例中 $n = 11$ ，要发送的 n 个数是 $[1, 10, 11]$ 。

首先我们把顺序调整成 $[10, 11, 1]$ 。

然后按照题目描述中的方法发送即可，共需要发送1110110位。不存在更优的方案。

2.7 数据规模与约定

对于20%的数据， $0 \leq n < 10$ 。

对于40%的数据， $0 \leq n < 100$ 。

对于60%的数据， $0 \leq n < 1000$ 。

对于80%的数据， $0 \leq n < 1000\ 0000$ 。

对于100%的数据， $0 \leq n < 10\ 0000\ 0000$ ，所有要发送的数都是随机生成的。

3 木乃伊危机

3.1 题目描述

在NOI 2106上的一次游览中，你碰到了一个埃及古墓。

不幸的是，你打开了坟墓之后，才发现这是一个坏主意：突然之间，原本空无一物的沙漠上已经爬满了暴躁的木乃伊。（如果你也沉睡几千年而突然被惊醒，你也会变得如此暴躁的。）

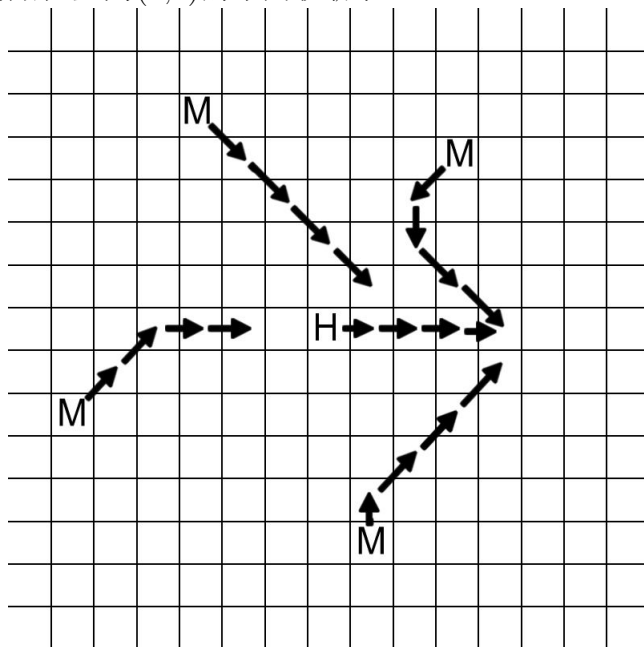
面对这一大堆疯狂的木乃伊，你唯一的机会就是试图在他们抓到你之前逃跑。问题是：假如你与木乃伊永不疲倦，那么经过多长时间你会被木

乃伊抓到?

我们把沙漠看成一个正方形的网格，你与木乃伊轮流移动。轮到你时，你可以移动到相邻的8个格子之一，或者站着不动。轮到木乃伊时，每个木乃伊会移动到其相邻的8个格子之一，使得他与你的欧几里得距离尽量小（假设你与木乃伊都站在格子的中心位置）。允许多个木乃伊同时占据同一个格子。

在每个单位时间内，你先做出移动，然后木乃伊做出移动。如果你与任何一个木乃伊站在同一位置，你会被抓住。当然，你试图尽量长时间避免被抓住。经过多少单位时间你会被抓住呢？

下图描述了你被4个木乃伊追逐的例子。H代表你的初始位置，而M代表木乃伊的初始位置。以你的初始位置为原点，则经过4个单位时间后，你被初始位置为(3,4)的木乃伊抓住。



3.2 输入格式

输入文件为mummy.in。

输入文件包含若干组数据。每组数据的第一行为一个数 n ($0 \leq n \leq 10^5$), 表示沙漠中木乃伊的个数。接下来 n 行, 每行两个整数 x y , 表示初始时在 (x, y) 有一个木乃伊。 x, y 的绝对值均不超过 10^6 。你的初始位置是 $(0, 0)$, 保证一开始这里没有木乃伊。

输入文件以一行-1结束。

3.3 输出格式

输出文件为mummy.out。

对于每组测试数据, 输出一行, 包括它的编号和被抓住经过的最长时间 (即你做出决策的次数); 或输出"never", 如果你有办法永远不被抓住。

请以样例输出的格式输出数据。

3.4 样例输入

```
4
-3 5
3 4
-6 -2
1 -5
1
0 -1
-1
```

3.5 样例输出

```
Case 1: 4
Case 2: never
```


3.6 数据规模与约定

对于20%的数据， $n \leq 100$;

对于50%的数据， $n \leq 1000$;

对于100%的数据， $n \leq 10^5$;

对于所有数据，每个输入文件只有1组数据，但是依然以-1结尾，并且你仍然需要输出“Case 1: ”。