

NOIP2023 模拟赛

时间：2023 年 9 月 28 日

题目名称	剧场	马	球衣	填数游戏
题目类型	传统题	传统题	传统题	传统题
目录	theatre	horse	soccer	number
可执行文件名	theatre	horse	soccer	number
输入文件名	theatre.in	horse.in	soccer.in	number.in
输出文件名	theatre.out	horse.out	soccer.out	number.out
每个测试点时限	1 秒	1 秒	1 秒	2 秒
内存限制	256 MB	256 MB	256 MB	512 MB
子任务数目	3	10	6	20
测试点是否等分	否	是	否	是

提交源程序文件名

对于 C++ 语言	theatre.cpp	horse.cpp	soccer.cpp	number.cpp
-----------	-------------	-----------	------------	------------

编译选项

对于 C++ 语言	-O2 -std=c++14
-----------	----------------

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. C/C++ 中函数 `main()` 的返回值类型必须是 `int`，程序正常结束时的返回值必须是 0。
3. 提交的程序代码文件的放置位置请参考各省的具体要求。
4. 因违反以上三点而出现的错误或问题，申诉时一律不予受理。
5. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
6. 选手提交的程序源文件必须不大于 100KB。
7. 程序可使用的栈空间内存限制与题目的内存限制一致。
8. 只提供 Linux 格式附加样例文件。
9. 评测在当前最新公布的 NOI Linux 下进行，各语言的编译器版本以此为准。

剧场 (theatre)

【题目描述】

魔都是一座高度现代化的城市，道路可以看作只有横向道路和竖向道路。因此我们视魔都城内两点的距离为他们的曼哈顿距离。魔都城的大小为 C, H ，即城内点的横坐标范围在 $[1, C]$ 之间，纵坐标范围在 $[1, H]$ 之间。

现在，魔都城内有 n 座主要建筑，他们的坐标分别为 $(x1_i, y1_i)$ ，保证坐标是整数。

市政府想在城市内修建一个剧场！该剧场有 m 个可能的选址 $(x2_i, y2_i)$ ，坐标同样是整数。你作为城市的规划员，你希望该剧场到所有主要建筑间距离的**最大值最小**。请你计算出这个最小值和对应的选址编号。

【输入格式】

从文件 *theatre.in* 中读入数据。

第一行两个整数 C, H 。表示魔都城的大小。

下一行一个整数 n ，表示主要建筑的数量。紧接的 n 行，每行两个整数 $x1_i, y1_i$ 表示主要城市坐标。

下一行一个整数 m ，表示选址的数量。接下来 m 行，一行两个整数 $x2_i, y2_i$ 表示选址的坐标。

【输出格式】

输出到文件 *theatre.out* 中。

第一行一个整数，表示剧场到主要城市间最小的最大值。

第二行一个整数，表示剧场选址的编号。标号按输入顺序从 1 到 m 编号。

【样例 1 输入】

```
1 10 10
2 2
3 1 1
4 3 3
5 2
6 1 10
7 4 4
```

【样例 1 输出】

```
1 6
2 2
```

【数据范围】

对于所有数据，满足 $C, H \leq 10^9, n \leq 10^5, m \leq 10^5$ 。

子任务编号	n	m	分值
1	$n = 1$	$m \leq 10^5$	10
2	$n \leq 5 \times 10^3$	$m \leq 5 \times 10^3$	30
3	$n \leq 10^5$	$m \leq 10^5$	60

马 (horse)

【题目描述】

有一个 $4 \times n$ 的棋盘，你初始可以在任意一个格子放置一个马，然后移动 $4n - 1$ 次，问能否不重不漏经过所有格子。如果可以，给出构造。

注：马走日。

【输入格式】

从文件 *horse.in* 中读入数据。

第一行一个正整数 T ，表示数据组数。

每组数据一行一个正整数 n 。

【输出格式】

输出到文件 *horse.out* 中。

对于每组数据：

如果有解，第一行输出 *Yes*，并在接下来 $4n$ 行，每行输出两个数 $x, y (1 \leq x \leq 4, 1 \leq y \leq n)$ ，表示依次经过的点的坐标。

如果无解，输出 *No*。

【样例 1 输入】

```
1 2
2 2
3 3
```

【样例 1 输出】

```
1 No
2 Yes
3 1 2
4 3 3
5 4 1
6 2 2
7 4 3
```

```
8 3 1
9 2 3
10 1 1
11 3 2
12 1 3
13 2 1
14 4 2
```

【数据范围】

对于 30% 的数据, $n \leq 5$ 。

对于 50% 的数据, $n \leq 10$ 。

对于 70% 的数据, $n \leq 1000$ 。

对于 100% 的数据, $n \leq 100000, T \leq 5$ 。

球衣 (soccer)

【题目描述】

球衣有 n 个颜色块需要被染色。球衣设计师正在思考该如何安排红蓝色。他的初步想法是以红色为底色, 在上面覆盖若干个连续的蓝色块。特别的, 你可以不用蓝色覆盖任何一个颜色块, 也可以用蓝色覆盖所有颜色块。他只确定了这些蓝色块的长度和相对位置, 而没有确定他们的具体位置。

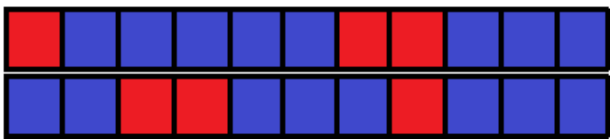
具体的, 他用数组 p 表示他的一个想法, 要求从左到右第 i 个极大连通蓝色块的大小恰好为 p_i 。容易发现, 这样的想法对应了多个可行的设计方案。

由于设计师没有确定想法 p 的最终设计方案, 他只记录下来了在想法 p 对应的所有设计方案中, 一定会被染成蓝色的颜色块的位置。

举例来说, 若想法 p 为 $\{3, 2, 3\}$, 那么它对应的设计方案可能是:



而以下的设计方案不可能是 $\{3, 2, 3\}$ 对应的设计方案:



第一种对应的方案是 $\{5, 3\}$, 第二种对应的方案是 $\{2, 3, 3\}$ 。在所有 p 对应的设计方案中, 一定会被染成蓝色的位置是:



而设计师有健忘症, 第二天醒来的时候他只记得在想法 p 对应的所有设计方案中, 所有一一定会被染成蓝色的颜色块的位置。为了让他的工作不白费, 他希望你能帮他还原出一种可能的设计方案, 或者告诉设计师他的记忆有问题, 即不存在这样的方案。

【输入格式】

从文件 `soccer.in` 中读入数据。

第一行一个正整数 n , 表示需要被染色的颜色块数量。

第二行一个长为 n 的 01 串, 其中 0 表示这个位置不一定被染成蓝色, 1 表示这个位置一定被染成蓝色。

【输出格式】

输出到文件 *soccer.out* 中。

如果不存在这样的想法 p , 输出一行一个字符串 No。

否则第一行输出一个字符串 Yes, 第二行输出一个长为 n 的字符串, 只包含字符 R 和 B, 表示一种合法的想法 p 对应的设计方案, 其中 R 表示红色, B 表示蓝色。如果有多种可能的设计方案, 输出任意一个即可。

【样例 1 输入】

```
1 11
2 01100100110
```

【样例 1 输出】

```
1 Yes
2 RBBBBRBBRBBB
```

【样例 1 解释】

一种可能的想法 p 是 $\{3, 2, 3\}$, 设计方案详见题目描述里的图片。

【样例 2 输入】

```
1 10
2 0000000000
```

【样例 2 输出】

```
1 Yes
2 RRRRRBRBRR
```

【样例 2 解释】

一种可能的想法 p 是 $\{1,1\}$, 样例输出给出了这个 p 对应的一个设计方案。当然, $p = \emptyset$ 也是合法的, 也就是说, 答案也可能为 RRRRRRRRRR。还有很多其他的合法方案, 这里就不一一列举了。

【样例 3】

见选手目录下的 `soccer/soccer3.in` 与 `soccer/soccer3.ans`。
该样例满足子任务 3 的限制。

【样例 4】

见选手目录下的 `soccer/soccer4.in` 与 `soccer/soccer4.ans`。
该样例满足子任务 6 的限制。

【提示】

由于不可抗力, 不提供设计方案的构造, 选手可以通过 `checker` 校验答案的正确性。
本题评测采用 `Special Judge`。

为了方便选手测试, 在附件中有 `checker.exe` 文件, 选手可以使用它校验自己的输出文件。`checker` 的使用方式为: `checker <inputfile> <outputfile> <answerfile>`, 参数依次表示输入文件、输出文件和答案文件。

提示: `checker` 只读取了 `answerfile` 的第一行, 也就是 `Yes` 或 `No`。对于答案是 `Yes` 的输入数据, 你只需要传入一个包含 `Yes` 的 `answerfile`, 即可检验你的 `outputfile` 的构造是否正确。

【数据范围】

子任务编号	$n \leq$	特殊性质	分值
1	15	无	15
2	100	无	15
3	3000	无	20
4	10^6	1 的个数 ≤ 1000	10
5	10^6	0 的个数 ≤ 1000	10
6	5×10^6	无	30

填数游戏 (number)

【题目描述】

小 L 是一个爱摸鱼的孩子，这天，小 L 玩起了填数游戏。

具体的，有 n 个格子，排成一行，编号分别为 $1 \sim n$ ，小 L 需要在每个格子中填入 $1 \sim k$ 中的一个数，对于最终的结果，令 a_i 为第 i 个格子中填入的数，那么代价为：

$$\max_{1 \leq i < j \leq n} (f[i][a_i] + f[j][a_j] + X \times |a_i - a_j|)$$

其中 $f[1 \cdots n][1 \cdots k]$ 和 X 已经提前给出。

小 L 想最小化代价，可是他由于天天摸鱼，所以不知道该怎么做。于是找到的聪明的你，希望你求出最小代价。

同时，他想知道构造方案，所以请你在此基础上构造出一组代价最小的方案，如果方案不唯一，任意输出一组即可。

【输入格式】

从文件 *number.in* 中读入数据。

第一行三个正整数， n, k, X 。

接下来 n 行，每行 k 个正整数，分别是 $f[i][1 \cdots k]$ 。

【输出格式】

输出到文件 *number.out* 中。

第一行一个数，表示最小的代价。

第二行 n 个数 $a_1 \cdots a_n$ ，表示每一个格子中填入的数。

【样例 1 输入】

```
1 2 2 1
2 2 4
3 4 2
```

【样例 1 输出】

```
1 5
2 1 2
```

【样例 2 输入】

```
1 3 3 2
2 1 3 4
3 3 3 4
4 4 5 1
```

【样例 2 输出】

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

【数据范围】

数据编号	$n \leq$	$k \leq$	特殊性质
1, 2	10	3	无
3, 4	10^3	2	无
5 ~ 8	10^5	2	无
9, 10	10^3	3	有
11 ~ 13	10^5	3	有
14 ~ 16	10^3	20	无
17 ~ 20	10^5	20	无

特殊性质：对于任意 $f[i][j](1 \leq i \leq n, 1 \leq j \leq k)$ ，均有 $f[i][j] > X$ ，并且保证 $k = 3$ 。
对于所有数据，保证 $2 \leq n \leq 10^5$ ， $2 \leq k \leq 20$ ， $1 \leq f[i][j], X \leq 10^8$ 。
本题输入量较大，请不要使用过慢的读入方式。