

# NOIP2023 模拟赛

时间：2023 年 7 月 7 日

## 一、题目概况

题目名称	骰子游戏	裁剪彩带	挑战 NPC	无聊的问题
题目类型	传统型	传统型	传统型	传统型
目录	dice	cut	hamilton	boring
可执行文件名	dice	cut	hamilton	boring
输入文件名	dice.in	cut.in	hamilton.in	boring.in
输出文件名	dice.out	cut.out	hamilton.out	boring.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒	3.0 秒
内存限制	256 MiB	256 MiB	256 MiB	512 MiB
子任务数目	3	20	20	5
测试点是否等分	否	是	是	否

## 二、提交源程序文件名

对于 C++ 语言	dice.cpp	cut.cpp	hamilton.cpp	boring.cpp
-----------	----------	---------	--------------	------------

## 三、编译选项

对于 C++ 语言	-lm -O2 -std=c++17
-----------	--------------------

## 四、注意事项

1. 选手提交的源程序请直接放在个人目录下，无需建立子文件夹。
2. 文件名（程序名和输入输出文件名）必须使用英文小写。
3. C/C++ 中函数 main() 的返回类型必须是 int，程序正常结束时的返回值必须是 0。
4. 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）
5. 程序可使用的栈内存空间限制与题目的内存限制一致。

# 骰子游戏 (dice)

## 【题目描述】

给你两个正整数  $n$  和  $k$ ，你需要构造  $n$  个正立方体骰子，每个骰子的每个面上都写有一个整数，满足：

- 写在骰子每个面上的每个整数都在  $[0, 10^6]$  范围内。
- 对于每个骰子，其六个面上写有的数字两两不同。
- 同时投出这  $n$  个骰子，所有骰子朝上面上写有数字的异或和都是  $k$  的倍数。

如果能构造出这样的  $n$  个骰子，请先输出 ‘Yes’，随后输出你的构造。

如果无法构造出这样的  $n$  个骰子，请输出 ‘No’。

## 【输入格式】

输入共一行，两个正整数，表示  $n$  与  $k$ ，含义见题目描述。

## 【输出格式】

输出共  $n + 1$  或 1 行。

第一行一个字符串 ‘Yes’ 或 ‘No’，表示是否存在这样的构造。

如果第一行输出 ‘Yes’，那么接下来  $n$  行每行六个整数。第  $i + 1$  行的六个整数表示写在第  $i$  个骰子六个面上的数。

## 【样例 1 输入】

3 2

## 【样例 1 输出】

Yes

1 3 5 7 9 11

3 5 7 9 11 2023

0 2 4 6 100000 10

## 【数据范围】

本题采用捆绑测试。

对于所有的数据，满足  $1 \leq n \leq 100, 2 \leq d \leq 60$ 。

- Subtask 1 (10pts)，满足  $n = 1$ 。
- Subtask 2 (20pts)，满足  $d = 2$ 。
- Subtask 3 (70pts)，无特殊限制。

# 裁剪彩带 (cut)

## 【题目描述】

你有一条长度为  $n$  的彩带，彩带的每个长度为 1 的区间都有一个颜色，从左到右的  $i$  长度位置的颜色为  $c_i$ 。

现在你要将彩带剪开，剪开后的一部分彩带的美丽值为这一部分所有颜色的 mex，一种剪彩带的方式的美丽度为剪开后每一部分的美丽值之和。

现在，你需要计算出美丽度最大的裁剪方案的美丽度是多少，并给出一种裁剪方案。

一个集合  $S$  的 mex 定义为最小的没有在  $S$  中出现的非负整数。

## 【输入格式】

第一行一个整数  $n$ ，表示彩带的长度。

第二行  $n$  个整数，表示彩带上每个位置的颜色。

## 【输出格式】

第一行两个整数  $s, k$ ，分别表示最大的美丽度与裁剪成的段数。

第二行  $k$  个整数，第  $i$  个整数表示裁剪后的第  $i$  段的起始位置。你需要保证这  $k$  个整数递增排列。

## 【样例 1 输入】

```
7
1 0 2 5 3 4 0
```

## 【样例 1 输出】

```
7 2
1 7
```

## 【样例 2 输入】

```
5
1 2 0 1 2
```

## 【样例 2 输出】

```
3 3
1 4 5
```

**【样例 3,4】**

见下发文件中的 *cut3/4.in* 与 *cut3/4.out*。

**【数据范围】**

对于所有的数据，满足  $1 \leq n \leq 5 \times 10^5$ ， $0 \leq a_i \leq 20$ 。

- 对于 20% 的数据，满足  $n \leq 10^3$ 。
- 对于另外 10% 的数据，满足  $a_i \leq 1$ 。

# 挑战 NPC (hamilton)

## 【题目背景】

c\*\*\*6 发明了一个在多项式复杂度内解决哈密顿回路的算法，现在他来考考你。但是他考虑到你还没有这么强，于是给出了这道题的弱化版。

## 【题目描述】

定义有向图上的两个回路不交，当且仅当这两个回路不经过公共点或公共边。

给你  $n$  个点， $m$  条边的有向图，求出若干两两不交的简单回路的并，使得经过每个点恰好一次。或报告无解。

如果有解，输出任意一个即可。

## 【输入格式】

输入数据共  $m + 1$  行。

第一行两个数  $n, m$ ，表示有向图的点数和边数。点的编号从 1 到  $n$ 。

接下来  $m$  行，每行两个数  $a_i, b_i$ ，表示一条从  $a_i$  到  $b_i$  的有向边，记作  $a_i \rightarrow b_i$ 。

## 【输出格式】

输出数据若干行。

如果无解，仅输出一行 'NO'。否则第一行输出 'YES'。

之后每行一个数  $k_i$ ，接下来  $k_i$  个数  $c_{i,1}, c_{i,2}, \dots, c_{i,k_i}$ ，表示回路  $c_{i,1} \rightarrow c_{i,2} \rightarrow \dots \rightarrow c_{i,k_i} \rightarrow c_{i,1}$ 。

你需要保证每个点总共出现恰好一次。

## 【样例 1 输入】

```
6 16
1 3
1 6
2 3
2 4
2 6
3 1
3 6
4 2
4 3
4 5
4 6
```

5 2  
5 6  
6 1  
6 3  
6 5

**【样例 1 输出】**

YES  
6 1 6 5 2 4 3

**【样例 2,3】**

见下发文件中的 *hamilton2/3.in* 与 *hamilton2/3.out*。

**【数据范围】**

对于所有的数据， $1 < n \leq 10^5$ ， $n \leq m \leq 2 \times 10^5$ 。

- 对于 40% 的数据， $1 < n \leq 20$ ， $n \leq m \leq 50$ 。
- 对于 80% 的数据， $1 < n \leq 10^3$ ， $n \leq m \leq 2 \times 10^3$ 。

# 无聊的问题 (boring)

## 【题目描述】

有一个长度为  $n$  的序列，每个数为  $a_i$ 。

现在有  $q$  个询问，每个询问先给出一个数  $k$ ，然后给出一些区间  $l_i, r_i$  求：

$$\text{lcm}_{1 \leq x \leq k} (\text{lcm}_{l_x \leq i \leq r_x} a_i) \bmod 998244353$$

## 【输入格式】

先给出  $n, q$ ,

接下来  $n$  个数，表示序列  $a_i$ 。

接下来  $q$  行，每行先是一个  $k$ ，然后接下来  $2k$  个数，表示  $[l_1, r_1], [l_2, r_2], \dots, [l_k, r_k]$ 。

## 【输出格式】

输出  $q$  行，每行一个答案。

## 【样例 1 输入】

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

## 【样例 1 输出】

```
21
252
1260
14
```

## 【样例 2,3】

见下发文件中的 *bornig2/3.in* 与 *boring2/3.out*。

### 【数据范围】

本题采用捆绑测试。

对于所有的数据，满足  $1 \leq n, q, a_i \leq 10^5, 1 \leq \sum k \leq 2.2 \times 10^5$ 。

- Subtask 1 (10pts)，满足  $n, m, a_i \leq 200, \sum k \leq 2.2 \times 540$ 。
- Subtask 2 (20pts)，满足  $n, m, a_i \leq 2000, \sum k \leq 4400$ 。
- Subtask 3 (20pts)，满足  $n, m, a_i \leq 10000, \sum k \leq 22000$ 。
- Subtask 4 (20pts)，满足  $a_i \leq 500$ 。
- Subtask 5 (30pts)，无特殊限制。