

ZXOI-3

Fido_Pigeon

时间：2023 年 6 月 28 日 13:40 ~ 15:40

题目名称	二分哥	匹配	质数
题目类型	传统型	传统型	Special Judge
目录	binary	match	prime
可执行文件名	binary	match	prime
输入文件名	binary.in	match.in	prime.in
输出文件名	binary.out	match.out	prime.out
每个测试点时限	2.0 秒	1.0 秒	5.0 秒
内存限制	512 MiB	512 MiB	512 MiB
测试点数目	20	25	4
测试点是否等分	是	是	否

提交源程序文件名

对于 C++ 语言	binary.cpp	match.cpp	prime.cpp
-----------	------------	-----------	-----------

编译选项

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

注意事项

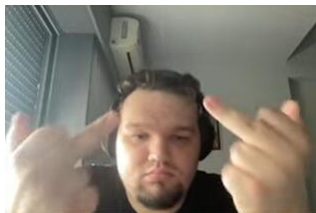
1. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
2. C++ 中函数 main() 的返回值类型必须是 int，值必须为 0。
3. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
4. 若无特殊说明，输入文件中同一行内的多个整数、浮点数、字符串等均使用一个空格进行分隔。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端下可使用命令 `ulimit -s unlimited` 将栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 评测在当前最新公布的 NOI Linux 下进行，使用 LemonLime 进行评测。
9. 我们保证题目背景与题目没有任何联系，你可以跳过题目背景的阅读。
10. 本次比赛部分测试点输入量较大，使用标准 cin 流读入可能很慢，您可以在程序开头使用 `ios::sync_with_stdio(false);cin.tie(0);cout.tie(0);` 来加快读入，请注意，在使用上述方式加快读入后，你将不能混用 C++ 风格 IO 和 C 风格 IO。

11. 由于出题人很菜，数据可能会很弱，请大家小声 **AK**，同时如果你认为有程序获得了不应有的分数，你可以联系出题人加强数据。
12. 由于出题人主观因素等，题目不一定按客观难度排序。
13. 本次比赛由不知道什么赞助商提供了不知道什么支持。
- 14.

二分哥 (binary)

【题目背景】

这是二分哥 `Um_nik`。



【题目描述】

`Um_nik` 是一个很强的 Codeforces 选手。

`Petr` 也是一个很强的 Codeforces 选手。

对于某个排列，我们定义一次“交换”为选择两个不同的位置将他们交换。

他们两个人各自拿到一个长度为 n 的初始升序的排列，随后 `Um_nik` 会将这个排列“交换” $7n + 1$ 次，而 `Petr` 会将这个排列“交换” $3n$ 次。

给你一个进行若干次交换后的排列，你需要判断这个排列是由谁进行操作得到的。

【输入格式】

从文件 `binary.in` 中读入数据。

本题单测试点内有多组测试数据。

第一行一个正整数 T ，表示测试数据组数。

对于每组测试数据：

第一行一个正整数 n, op ，表示排列长度和序列生成方式。

若 $op = 0$ ，则第二行 n 个正整数，表示被操作后的排列。

若 $op = 1$ ，则第二行两个正整数 sda, sdb ，表示这个被操作后的排列将由给定的数据生成器生成。

【输出格式】

输出到文件 `binary.out` 中。

对于每组测试数据，输出一行 `Um_nik` 或 `Petr`，表示这个排列是由谁操作得到的。

请注意 `Um_nik` 中的下划线为英文半角下划线，为了防止渲染问题造成的不必要影响，我们推荐你手打一遍这个名字。

【样例 1 输入】

```
1 5
2 5 0
3 5 4 3 2 1
4 6 0
5 1 6 2 5 4 3
6 3 0
7 3 2 1
8 7 0
9 7 3 1 6 5 2 4
10 10 1
11 23 34
```

【样例 1 输出】

```
1 Um_nik
2 Um_nik
3 Petr
4 Petr
5 Petr
```

【样例 1 解释】

第五组数据生成的排列为：

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

【样例 2】

见选手目录下的 *binary/binary2.in* 与 *binary/binary2.ans*。

【数据范围】

测试点	$n \leq$
1~3	5
4~8	10^3
9~17	10^5
18~20	2.5×10^6

对于所有数据, 满足 $1 \leq T \leq 10$, $1 \leq n \leq 2.5 \times 10^6$, $0 \leq sda, sdb < 2^{32}$ 。
对于测试点 18 ~ 20, 额外满足: $op = 1$ 。

【数据生成方式】

我们在选手文件夹下发了一个数据生成器的源文件, 打开后你应当看到如下内容:

```
1 #include <vector>
2 struct Generator{
3     ...
4     std::vector<int> get_permutation(int n,int a_,int b_)
5     {
6         ...
7     }
8 }gen;
```

对于 $op = 1$ 的测试点, 你需要将这份代码粘贴至主函数之前。

随后调用 `gen.get_permutation(n,sda,sdb)` (n, sda, sdb 都由输入给出, 详见输入格式), 生成器将返回一个长度为 n 的排列并以 `std::vector<int>` 为容器存储。

请注意, 经过我们的测试, 对于测试点 18 ~ 20, 数据生成器生成所有数据的时间约为 0.8 秒, 请注意程序运行速度。

匹配 (match)

【题目背景】

I'm losing my innocence
From the shadow of my mind
Can you handle it? Fire in the sky
...
... Don't fade away
... I made mistakes
... Don't let me fade
She flying to the moon and back
She flying to the moon and back
Moon oh moon, Moon oh moon
Five, Four, Three, Two, One
I'll lift off

【题目描述】

哈娜有一棵树。

哈娜喜欢一些稳定的匹配，因此哈娜希望这棵树的最大匹配唯一。

显然有些情况下原来的树可能不满足这个性质，因此哈娜想在树上断开几条边，使得断边之后的树（其实应该叫森林了）最大匹配唯一。

你需要帮助哈娜求出有多少种删边方案使得最大匹配唯一。

称两种删边方案不同，当且仅当存在某条边，他在两种方案中一个被保留，一个被删除。

答案对 998244353 取模。

【输入格式】

从文件 *match.in* 中读入数据。

第一行一个正整数 n ，表示树的点数。

接下来 $n - 1$ 行，每行两个数 u, v ，表示树上存在一条从 u 到 v 的双向边。

【输出格式】

输出到文件 *match.out* 中。

输出一个数，表示方案数，对 998244353 取模。

【样例 1 输入】

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

【样例 1 输出】

```
1 4
```

【样例 1 解释】

有以下断边方式：

- 断掉 (1,2) 和 (1,3)。
- 断掉 (1,2) 和 (1,4)。
- 断掉 (1,3) 和 (1,4)。
- 断掉所有边。

【样例 2 输入】

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

【样例 2 输出】

```
1 6
```

【样例 2 解释】

有以下断边方案：

- 一条边都不断。
- 断掉 (1,2) 和 (2,3)。
- 断掉 (1,2) 和 (3,4)。
- 断掉 (2,3) 和 (3,4)。

- 断掉 (2,3)。
- 断掉所有边。

【样例 3 输入】

1 1

【样例 3 输出】

1 1

【样例 4】

见选手目录下的 *match/match4.in* 与 *match/match4.ans*。

【样例 5】

见选手目录下的 *match/match5.in* 与 *match/match5.ans*。

【数据范围】

测试点	$n \leq$	特殊性质
1~4	10	无
5~8	2×10^3	A
9~14		无
15~20	3×10^5	B
21~25	3×10^5	无

特殊性质 A：对于每个点，其度数不超过 10。

特殊性质 B：对于第 i 条边，有 $u = i, v = i + 1$ 。

对于所有数据，满足： $1 \leq n \leq 3 \times 10^5$ 。

不知道这波腾讯调出来了没有。

质数 (prime)

【题目背景】

Such a destiny was not desired

In this place

I never wished to see that

Side of you

And raise

Different colors for one another

Never notice me

Never call my name

With your voice

Let me forget everything

And leave no trace of you

What am I carrying

Out of this horizon of despair

If it does not you

I can hurt anyone

In the name of justice

Your shadow

Gets in the way of my light

And Testify

【题目描述】

给你一个正整数 n ，你需要找出一个区间 $[l, r]$ ，满足 $[l, r]$ 内所有质数的和等于 n ，若无解，请输出 **NIE**。

本题采用了 **Special Judge**，你的答案正确当且仅当：

- 若输入无解，你的程序只输出了 **NIE**。
- 若输入有解，你的程序输出了一个区间 $[l, r]$ ，且区间内所有质数的和等于 n 。

【输入格式】

从文件 *prime.in* 中读入数据。

一行一个正整数 n 。

【输出格式】

输出到文件 *prime.out* 中。

如果有满足条件的区间，则输出两个数 l, r 即区间的左右端点。

否则输出 NIE。

【样例 1 输入】

1 15

【样例 1 输出】

1 3 7

【样例 1 解释】

$3 + 5 + 7 = 15$ 。

【样例 2 输入】

1 9992

【样例 2 输出】

1 4993 4999

【样例 3 输入】

1 100000000

【样例 3 输出】

1 NIE

【样例 4 输入】

```
1 1000000007
```

【样例 4 输出】

```
1 1000000007 1000000007
```

【样例 5 输入】

```
1 99999999996
```

【样例 5 输出】

```
1 295693 1693067
```

【数据范围】

本题采用捆绑测试。

子任务	$n \leq$	得分
1	10^4	15
2	10^8	20
3	2×10^9	40
4	10^{11}	25

对于所有数据，满足 $1 \leq n \leq 10^{11}$ 。

【提示】

我们在选手文件夹中下发了一个 `sum.cpp` 的文件，编译后，你可以输入两个正整数 l, r ，程序会返回 $[l, r]$ 内的质数和，请注意，当这个和过大时，程序会输出 **Out of Bounds!**。