

FJWC2019 练习赛

Lsk,Timsei

试题数目	3
时间限制	3.5h

题目名称	循环流	整除分块	森林
题目类型	传统	传统	传统
目录	flow	mex	forest
可执行文件名	flow	mex	forest
输入文件名	flow.in	mex.in	forest.in
输出文件名	flow.out	mex.out	forest.out
单测试点时限	1s	2s	3s
单测试点内存限制	1024MB	1024MB	1024MB
子任务(或测试点)数	5	7	6
分值	100	100	100
C++语言编译选项	-O2 -lm	-O2 -lm	-O2 -lm
C 语言编译选项	-O2 -lm	-O2 -lm	-O2 -lm
Pascal 语言编译选项	-O2	-O2	-O2

提示:

1. 每题时间限制都在标准程序运行时间的两倍以上，如果你发现你的程序运行超过时限，请你更多考虑优化算法的时间复杂度而非常数因子。
2. 每题都下发若干样例文件，你可以合理利用这些数据全面测试你的程序。
3. 由于时间仓促以及出题人水平有限，难免会有疏忽之处，如果发现试题中的各种问题请及时提出，希望大家能做题愉快

循环流

背景

你学习了网络流的相关知识，现在你要大显神通.....

题目描述

你曾经有一个循环流(每个点均满足流量平衡条件)，这个流网络上有 n 个点，且每条边的流量只有 1 或 2，可能有重边却没有自环。显然，由于它是一个流网络，它是一个弱连通图(将边视为无向边后为连通图)。遗憾的是你找不到这个流了，但你曾经记下了它流量为 1 的边的数量和流量为 2 的边的数量。由于这个图的点数有点多，你很有可能数错边数，因此你现在想知道存不存在这样一个流符合你记下的数据。

输入格式

从文件 flow.in 中读入数据；

第一行一个整数表示数据所属的测试点编号，其具体含义见测试点约束(若为 0 表示样例数据)；

第二行一个整数 T 表示数据组数；

接下来 T 行每行三个整数 n, a, b , 分别表示点数, 流量为 1 的边的数量和流量为 2 的边的数量。

输出格式

输出到文件 flow.out 中；

输出 T 行，每行一个为 0 或 1 的整数，1 表示存在一个合法流，0 表示不存在。

样例输入 1

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

样例输出 1

```
1
1
0
```

样例输入 2

见选手文件下的 sample_flow2.in。

注意这组样例没有样例输出，其正确输出中共有 74 个 1。

测试点约束

对于所有的测试数据，保证：

$$2 \leq n \leq 50$$

$$0 \leq a, b \leq 50$$

$$1 \leq T \leq 127449$$

子任务编号	分值	n, a, b
1	20	≤ 2
2	15	≤ 3
3	15	≤ 4
4	30	≤ 5
5	20	≤ 50

整除分块

背景

你正在认真做题。

你切穿了许多整除分块的水题,但喜欢深入思考的你更想知道和整除分块有关的一些更强的性质.....

题目描述

整除分块是一种针对向下取整除法的常见处理技巧,多用于解决一些数论问题。它基于一个重要结论:对于正整数 n , 定义无限数列 a_n :

$$a_{n,i} = \left\lfloor \frac{n}{i} \right\rfloor \quad (i \in \mathbb{N}^*)$$

可以证明数列 a_n 中不同的数的个数只有 $O(n^{0.5})$ 种。

但本题中我们不关心数列 a_n 中不同的数的个数。

你手算了几个数后发现很小的数基本都在数列 a_n 中出现过,因此你希望了解函数 f , 其中 $f(n)$ 表示数列 a_n 中最小的没有出现过的自然数。

为了证明你对函数 f 十分了解,我们给出了 T 组询问, 每组询问会给出 l, r , 查询表达式 $\sum_{l \leq k \leq r} f(k)$ 对 998244353 取模的值, 你需要快速完成回答。

输入格式

从文件 `mex.in` 中读入数据;

第一行一个正整数表示该数据的测试点编号, 其具体含义见测试点约束(若为 0 表示样例数据);

第二行一个正整数 T 表示询问数;

接下来 T 行每行 2 个正整数 l, r , 描述了一次询问, 表示这一次你需要回答表达式 $\sum_{l \leq k \leq r} f(k)$ 对 998244353 取模的值。

输出格式

输出到文件 `mex.out` 中;

输出 T 行, 每行一个在 $[0, 998244352]$ 中的整数, 表示这一次查询的答案。

样例输入 1

```
0
3
1 1
5 5
10 10
```

样例输出 1

2
3
4

样例解释

数列 a_1 中出现过的自然数集合为 $\{0,1\}$, $f(1)$ 为 2;

数列 a_5 中出现过的自然数集合为 $\{0,1,2,5\}$, $f(5)$ 为 3;

数列 a_{10} 中出现过的自然数集合为 $\{0,1,2,3,5,10\}$, $f(10)$ 为 4;

样例输入 2

见选手文件下的 sample_mex2.in。

样例输出 2

见选手文件下的 sample_mex2.ans。

测试点约束

对于所有的测试数据, 保证:

$$3 \leq T \leq 65536$$

$$1 \leq l \leq r \leq 10^{36}$$

测试点编号	分值	T	l, r
1	15	=65536	$\leq 10^5$
2	5		$\leq 2 \times 10^6$
3	17		$\leq 2 \times 10^7$
4	17	=16384	$\leq 10^{12}$
5	16	=4	$\leq 10^{18}$
6	20	= 65536	
7	10		$\leq 10^{36}$

提示

注意到第 7 个测试点中 l, r 的数据范围超过了 64 位整形表示的范围, 但评测系统上 C++ 语言可以支持 128 位带符号整数 `__int128` 类型, 你可以使用它存储 $[-2^{127}, 2^{127} - 1]$ 范围内的整数, 但该类型不能使用 `cin/cout/scanf/printf` 读入或输出

森林

背景

幻想世界里有一片森林，森林里自然有许多许多树.....

题目描述

我们定义对一棵树做二次变换的含义为：当以 1 号节点为根时，交换两个互相不为祖先的点的子树；

一棵树的权值为对它进行至多一次变换能得到的最大直径长度；

初始时你只有一个节点 1，你需要执行 $n-1$ 个操作，第 i 次操作会给出一个整数 x ，表示新加入第 $i+1$ 号点，并与第 x 号点连一条边。每次操作后输出当前的树的权值。

由于某些原因我们对数据进行了强制在线处理。

输入格式

从文件 forest.in 中读入数据；

第一行一个整数表示数据所属的子任务编号，其具体含义见子任务约束(若为 0 表示样例数据)；

第二行一个整数 n ，表示所有操作完成后树的点数；

接下来一行 $n-1$ 个整数，第 i 个整数 a_i 表示第 i 次操作前树的权值与这次操作的 x 值的异或值。

输出格式

输出到文件 forest.out 中；

输出 $n-1$ 行,表示每次操作之后的树的权值。

样例输入 1

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

样例输出 1

```
1
2
3
3
```

样例解释

加入的三条边依次为 (1,2), (1,3), (2,4), (1,5)。

样例输入 2

见选手文件下的 sample_forest2.in。

样例输出 2

见选手文件下的 sample_forest2.ans。

子任务约束

本题采用子任务方式评测，只有通过了一个子任务中的所有测试点才能得到该子任务的分值。

对于所有的测试数据，保证：

$$1 \leq n \leq 2 \times 10^5,$$

任意时刻树的形态合法

子任务编号	分值	n	特殊限制
1	13	≤ 50	无
2	20	$\leq 5 \times 10^3$	
3	20	$\leq 2 \times 10^5$	相连的点编号相差不超过 2
4	10		保证树以 1 为根的深度不超过 50
5	10		保证数据随机
6	27		无