

2024 NOIP模拟赛

by GDSY

题目名称	定向越野	字符串	数数	逃脱
题目类型	传统型	传统型	传统型(spj)	传统型
目录	walk	str	count	esc
可执行文件	walk	str	count	esc
输入文件名	walk.in	str.in	count.in	esc.in
输出文件名	walk.out	str.out	count.out	esc.out
每个测试点时限	1s	1s	5s	1s
内存限制	256MB	256MB	256MB	256MB
是否捆绑测试	否	否	是	否
提交文件名	walk.cpp	str.cpp	count.cpp	esc.cpp

编译选项 `-lm -std=c++14 -o2 -w1,--stack=998244353`

定向越野 (walk)

时空限制

时间限制：1000 ms

空间限制：256 MB

题目描述

你觉得定向越野太累了，于是你造了个机器人帮你跑。在二维平面上有 n 个目标点，你的机器人需要从 1 号点出发，以任意顺序经过所有的目标点，最后回到 1 号点处。

但是由于设计缺陷，你的机器人在出发后只能进行直角转弯。

现在，你需要计算你的机器人至少需要移动多少距离才能完成定向越野任务。

输入格式

从 `walk.in` 中读入数据。

第一行一个整数 n ，表示目标点数量。

接下来 n 行每行两个整数 x, y ，表示第 i 个目标点在平面直角坐标系上的坐标。

输出格式

输出到 `walk.out` 中。

可以证明“机器人移动的距离”的平方为分数。

请将该分数化简为最简分数。

一行两个整数，分别表示该分数的分子与分母。

样例组

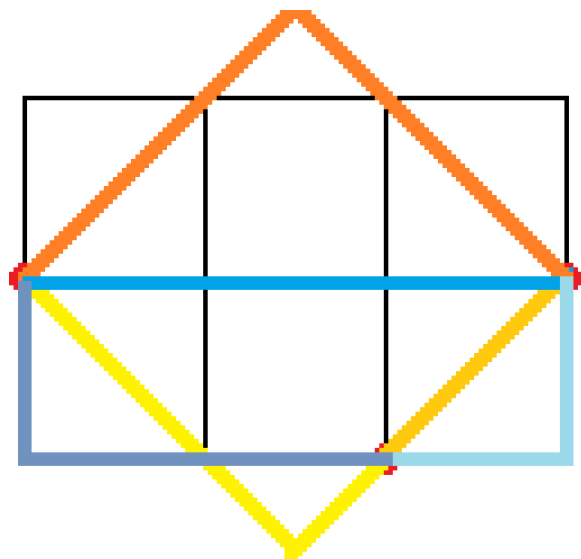
输入样例 1

1	3
2	0 1
3	2 0
4	3 1

输出样例 1

1	64 1
---	------

样例1解释



如图表示了两种可能的越野方案
其长度分别为 8 和 $6\sqrt{2}$

数据范围

$$2 \leq n \leq 12$$

$$0 \leq x, y \leq 5000$$

数据点编号	n
1 ~ 5	数据点编号 $\times 2$
6 ~ 10	12

字符串 (str)

时空限制

时间限制：1000 ms

空间限制：256 MB

题目描述

给定一个长度为 m 的串 s ，和两个权值数组 a, b 。

定义这个串的权值为 $\sum_{i=1}^m \sum_{j=1}^i a_j \times b_i \times \prod_{k=1}^{i-j+1} [s_k = s_{j-1+k}]$

注：这里 $[s_k = s_{j-1+k}]$ 表示判断，若为真则取值为 1，否则为 0

为了加大难度，改为给定长度为 n 的串 S 。对于每个 $m = 1, 2 \dots n$ ，你需要求 S 长度为 m 的前缀的权值

为了进一步加大难度，题目要求强制在线，你需要用长度为 m 的答案来解密 S 第 $m + 1$ 位的值

具体解密方式见输入格式

输入格式

从 `str.in` 中读入数据。

第一行两个整数 n, t ，表示串的总长与是否强制在线， $t = 0$ 表示不强制在线， $t = 1$ 表示强制在线。

第二行 n 个整数 a_i

第三行 n 个整数 b_i

第四行 n 个整数 S'_i ，用于解密字符串 S

设 ans_i 为长度为 i 时的答案，则 $S_{i+1} = (S'_{i+1} + t \times ans_i) \& 31$

其中 $\&$ 表示按位与

特别的定义 $ans_0 = 0$

输出格式

从 `str.in` 中读入数据。

n 行，每行一个整数，表示 ans_i

样例组

输入样例 1-1

```
1 7 0
2 1 2 3 4 5 6 7
3 7 6 5 4 3 2 1
4 1 2 1 3 1 2 1
```

输入样例 1-2

1	7	1
2	1	2 3 4 5 6 7
3	7	6 5 4 3 2 1
4	1	27 20 2 28 11 30

输出样例 1

1	7
2	13
3	33
4	37
5	55
6	67
7	80

数据范围

$1 \leq n \leq 10^6$

$t \in \{0, 1\}$

$0 \leq a_i, b_i \leq 10^3$

$0 \leq S'_i \leq 31$

数据点编号	n	t	S_i	a_i, b_i
1	$\leq 10^3$			
2	$\leq 10^4$	$= 0$		
3	$\leq 10^4$			
4		$= 0$	≤ 1 且保证等概率随机	
5		$= 0$	$= \log_2 lowbit(i)$	
6 ~ 7		$= 0$		
8 ~ 11				

注：其中数据点 4 和 5 的分数为 5 分；其余数据点分数为 10 分；共计 100 分

数数 (count)

时空限制

时间限制：5000 ms

空间限制：256 MB

题目描述

在一个长度为 k 的数列中，如果一个数的出现次数（严格）超过了 $\lfloor \frac{k}{3} \rfloor$ ，你就认为这个数是“优势巨大”的。

现在你需要维护一个长度为 n 的数列 A （下标从 1 开始），支持以下操作。

1. 给定 l, r, c ，将所有下标在 $[l, r]$ 的数加上 c 。即 $A_i = A_i + c, i \in [l, r]$
2. 给定 l, r, c ，将所有下标在 $[l, r]$ 的数改为 c 。即 $A_i = c, i \in [l, r]$
3. 给定 l, r ，请输出一个大小为 2 的集合，使其中包含子数列 $A_{[l, r]}$ 所有“优势巨大”的数

输入格式

从 `count.in` 中读入数据。

第一行两个整数 n, m ：分别表示数组长度和操作次数。

第二行 n 个整数 a_i ：表示数组的初始情况。

接下来 m 行，每行 4 个或 3 个整数 $x, l, r, (c)$ ：其中第一个整数 x 表示操作类型；接下来两个整数 l, r 表示这次操作考虑的下标区间；若操作类型为 1 或 2，接下来一个整数 c 表示操作参数。

输出格式

输出到 `count.out` 中。

对于每个询问，输出一行两个整数，表示你对“优势巨大”的数的猜测。

可以证明“优势巨大”的数最多有两个。

你输出的两个数的集合应当包含所有实际“优势巨大”的数。具体如下。

1. 如果实际没有“优势巨大”的数，你可以任意输出两个（int 范围内的）整数。
2. 如果实际上有且仅有一个“优势巨大”的数，它应当是你输出的两个数之一，你可以任意地输出另一个数。
3. 如果实际有两个“优势巨大”的数，你应当输出这两个数，可以以任意的顺序。

注：无论如何，你都应当输出两个（int 范围内的）整数，否则 `spj` 会识别错位并返回 `Wa`。

样例组

输入样例 1

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

输出样例 1-1

```
1 | 0 4
2 | 1 2
3 | 0 0
```

输出样例 1-2

```
1 | 4 0
2 | 1 2
3 | 998244353 1000000007
```

样例测试说明

下发文件中有 `checker.exe`，可以以命令 `checker.exe in.txt out.txt ans.txt` 调用。

例如：你的程序对样例输入 `count01.in` 的输出为 `count01.out`，样例答案为 `count01.ans`。你可以用 `checker.exe count01.in count01.out count01.ans` 命令来判断正误。

你可能希望自己造样例用于测试，答案文件格式如下：

- 对于每组询问，输出一行。其中第一个数为实际“优势巨大”的数的个数 k ，接下来 k 个数为实际“优势巨大”的数
- 注意行末不要有空格
- 注意文末不要有换行

数据范围

$1 \leq n, m \leq 500000$

$1 \leq l \leq r \leq n$

$|A_i|, |c| \leq 1000$

注：这里 $|A_i| \leq 1000$ 仅限制初值，不保证在操作后仍有 $|A_i| \leq 1000$ 。

具体每个测试点数据范围如下，留空表示无额外限制。

本题开启 subtask，下表每行表示一个 subtask。

保证每个 subtask 的数据点个数 ≤ 4

保证数据点不会根据选手更改

subtask 编号	n	m	其他限制	分值
1	≤ 1000	≤ 1000		10
2	≤ 5000	≤ 5000		20
3	≤ 50000	≤ 50000		20
4				50

（备注：请相信自己代码的常数，它真的足够小；你可以自己出一个极限数据测试一下时间）

逃脱 (esc)

时空限制

时间限制：1000 ms

空间限制：256 MB

题目描述

你在玩一款非对称双人对战游戏

游戏在一棵被称为“迷宫”的 n 个节点的树上进行。所有度数为 1 的节点被称为是“迷宫”的“出口”。

初始时游戏的“逃脱方”所操控的“逃脱者”会随机出生在“迷宫”的一个节点上，此时游戏双方都知道逃脱者的位置。随后，你作为“围堵方”，可以选择在地图的若干个出口处召唤“围堵者”。

之后双方轮流进行操作：

- 逃脱方操作时可以将逃脱者移动至相邻的一个节点
- 围堵方操作时可以将**每个**围堵者移动至其相邻的一个节点

任意时刻，如果“逃脱者”和某个“围堵者”在同一个节点上，则围堵方获胜；否则，如果“逃脱者”在任意出口处，则逃脱方获胜。

在游戏的全过程中，游戏双方均知晓所有角色所处位置和“迷宫”样式。

你想要写一份代码来计算你至少召唤多少个“围堵者”，才能保证获胜

由于你的电脑很卡，无法在游戏开始后再运行别的代码，所以你需要写一份代码，使其对于“逃脱方”的每一个出生点，都输出对应的结果。

输入格式

从 `esc.in` 中读入数据。

一行一个整数 n ，表示“迷宫”总节点个数。

接下来 $n - 1$ 行，每行两个整数 u, v ，表示节点 u 和节点 v 相连

保证输入的是一棵树

输出格式

输出到 `esc.out` 中。

输出一行 n 个整数，第 i 个数表示“逃脱者”出生在 i 号节点时，所需要的“围堵者”人数

样例组

输入样例 1

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


输出样例 1

1	2	3	3	1	1	1	1
---	---	---	---	---	---	---	---

数据范围

$1 \leq n \leq 7 \times 10^4$

保证输入的路径形成一棵树

数据点编号	n	特殊性质
1		保证输入的树形成一条链
2		保证叶子个数不超过 10
3 ~ 4	$\leq 10^4$	
5 ~ 10		