

NOI2025 模拟赛

题目名称	树	序列	栈
题目类型	传统型	传统型	传统型
可执行文件名	tree.exe	sequence.exe	stack.exe
输入文件名	tree.in	sequence.in	stack.in
输出文件名	tree.out	sequence.out	stack.out
每个测试点时限	1.0 秒	2.0 秒	1.0 秒
内存限制	512 MiB	512 MiB	512 MiB
测试点数目	10	20	20
测试点是否等分	是	是	是

提交源程序文件名

对于 C++ 语言	tree.cpp	sequence.cpp	stack.cpp
-----------	----------	--------------	-----------

编译选项

对于 C++ 语言	-O2 -std=c++14 -Wl,--stack=536870912
-----------	--------------------------------------

树 (tree)

【题目描述】

有两棵树，大小均为 n 。两棵树均以 1 号节点为根。

你可以进行如下操作：选择其中一棵树的一个非根节点 x ，将 x 的所有儿子和 x 的父亲连边，然后删去 x 以及与 x 连接的边。容易看出，一棵树经过操作后还是一棵树。

你想要使得两棵树相同。具体地，对于两棵树中编号相同的两个点，它们的父亲编号也应相同。求最小的操作次数。

注意这里的“相同”并非“同构”。

【输入格式】

从文件 *tree.in* 中读入数据。

输入的第一行包含一个正整数 n 。

接下来 $n - 1$ 行，每行包含两个正整数 u, v ，表示第一棵树中编号为 u 和 v 的点之间存在一条边。

接下来 $n - 1$ 行，每行包含两个正整数 u, v ，表示第二棵树中编号为 u 和 v 的点之间存在一条边。

【输出格式】

输出到文件 *tree.out* 中。

输出一行，包含一个整数，表示最小的操作次数。

【样例 1 输入】

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

【样例 1 输出】

14

【样例 2 输入】

12
21 2
32 1

【样例 2 输出】

10

【样例 3】

见选手目录下的 *tree/tree3.in* 与 *tree/tree3.ans*。

【样例 4】

见选手目录下的 *tree/tree4.in* 与 *tree/tree4.ans*。

【子任务】

对于所有数据，保证 $1 \leq n \leq 40$ 。保证给出的是两棵树。

测试点	$n \leq$
1, 2	10
3, 4	20
5, 6	30
7 ~ 10	40

序列 (sequence)

【题目描述】

给定一个长度为 n 的序列 a_1, a_2, \dots, a_n ， 和一个正整数 C 。求

$$\max_{S \subseteq \{1, 2, \dots, n\}} \left(\left(\sum_{1 \leq l \leq r \leq n} \prod_{i=l}^r [i \in S] \right) C - \sum_{i \in S} a_i \right)$$

这个问题太简单了，因此你需要支持 m 次单点修改。
但这样就太难了，因此修改是临时修改，不会保留。

【输入格式】

从文件 `sequence.in` 中读入数据。
输入的第一行包含三个整数 n, m, C 。
输入的第二行包含 n 个整数，表示序列 a 。
接下来 m 行，每行包含两个整数 x, y ，表示询问将 a_x 修改为 y 之后的答案。

【输出格式】

输出到文件 `sequence.out` 中。
输出 $m + 1$ 行，每行包含一个整数。
第一行表示所有修改开始之前，题中算式的值。
第 $i + 1$ 行表示如果执行第 i 次修改，题中算式的值会变为多少。

【样例 1 输入】

```
1 5 2 1
2 1 1 4 1 1
3 3 2
4 3 10
```

【样例 1 输出】

```
1 7
2 9
3 2
```

【样例 2 输入】

```
1 12 10 2
2 2 3 2 7 8 3 2 1 20 5 15 20
3 9 3
4 11 1
5 5 35
6 6 15
7 12 1
8 1 9
9 4 3
10 10 2
11 5 1
12 7 6
```

【样例 2 输出】

```
1 68
2 85
3 82
4 41
5 56
6 87
7 61
8 72
9 71
10 75
11 64
```

【样例 3】

见选手目录下的 *sequence/sequence3.in* 与 *sequence/sequence3.ans*。

【样例 4】

见选手目录下的 *sequence/sequence4.in* 与 *sequence/sequence4.ans*。

【样例 5】

见选手目录下的 `sequence/sequence5.in` 与 `sequence/sequence5.ans`。

【子任务】

对于所有数据，保证 $1 \leq n \leq 3 \times 10^5$ ， $0 \leq m \leq 3 \times 10^5$ ， $1 \leq C \leq 10^5$ ， $1 \leq x \leq n$ ， $0 \leq a_i, y \leq 10^9$ 。

测试点	$n \leq$	$m \leq$	特殊性质
1, 2	15	15	无
3 ~ 6	300	300	
7 ~ 10	2,000	2,000	
11, 12	3×10^5	0	
13, 14		3×10^5	$x \leq 10$
15, 16	10^5	10^5	无
17 ~ 20	3×10^5	3×10^5	

栈 (stack)

【题目描述】

有三个栈 S_1, S_2, S_3 。初始 S_1 中有 n 个数，自顶向下依次是 p_1, p_2, \dots, p_n ，保证 p 是一个排列。 S_2 和 S_3 初始为空。

你可以进行两种操作：

1. 选择一个正整数 $x \leq A$ ，将 S_1 栈顶的 x 个数作为一个整体弹出，并按原来的顺序放进 S_2 ；
2. 选择一个正整数 $x \leq B$ ，将 S_2 栈顶的 x 个数作为一个整体弹出，并按原来的顺序放进 S_3 。

其中 A 和 B 是给定的正整数。

你需要让最终 S_3 中的数自顶向下依次是 $1, 2, \dots, n$ 。构造一种合法的方案，或者输出无解。

【输入格式】

从文件 `stack.in` 中读入数据。

本题有多组测试数据。

输入的第一行包含一个正整数 T ，表示数据测试组数。接下来输入每组测试数据。

对于每组测试数据，第一行三个正整数 n, A, B ，第二行 n 个正整数 p_1, p_2, \dots, p_n 。含义见题目描述。

【输出格式】

输出到文件 `stack.out` 中。

对于每组测试数据：

如果无解，输出一行 `-1`。

否则，第一行输出一个整数 m ，表示你构造的方案的操作次数。

接下来 m 行，每行两个正整数 p, x ，表示执行第 p 种操作，选择的正整数是 x 。

你只需给出任意一种方案即可。

【样例 1 输入】

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

```
6 4 2 2
7 3 2 1 4
```

【样例 1 输出】

```
1 5
2 1 2
3 1 1
4 2 1
5 2 1
6 2 1
7 11
8 1 1
9 1 1
10 1 3
11 1 1
12 2 1
13 1 1
14 2 1
15 2 1
16 2 1
17 2 2
18 2 1
19 -1
```

【样例 2】

见选手目录下的 *stack/stack2.in* 与 *stack/stack2.ans*。

【子任务】

对于所有数据，保证 $n \geq 1$ ， $\sum n \leq 10^6$ 。保证 $1 \leq A, B \leq n$ 且 p 是排列。

测试点	$\sum n \leq$	$A = n$	$B = n$
1, 2	10	否	否
3 ~ 5	20		
6 ~ 8	2, 000	是	是
9, 10	10^6		否
11, 12	2, 000		
13	10^6		
14, 15	2, 000	否	是
16	10^6		否
17, 18	2, 000		
19, 20	10^6		

【提示】

本题输入输出量较大，请采用较为快速的输入输出方式。