

ZTOI R1 公开赛题目 PDF 版本

hi~ 这里是 ZTOI Round 1, 为了防止洛谷在比赛期间挂掉, 导致无法查看题目, 我们在这里给大家提供 PDF 版本的题目。

祝所有选手比赛顺利!

我们的题目列表:

# [†]	Type	Idea	Check	Data	Std	Solution
Z	传统	rui_er	ztrztr	rui_er	rui_er	Link
A	传统	幸存者	Sktic	幸存者	幸存者	Link
B	传统	Larunatreacy	Sktic	Larunatreacy	Larunatreacy	Link
C	传统	Larunatreacy	Sktic	Larunatreacy	Larunatreacy	Link
D	交互 [‡]	rui_er	幸存者	rui_er	rui_er	Link
E	传统	Larunatreacy	待定	Larunatreacy	Larunatreacy	Link
F	传统	Larunatreacy	hitroad	Larunatreacy	Larunatreacy	Link

「ZTOI R1」植树

题目描述

小 R 是一个提倡环保的女孩子。为了践行她的环保理念, 她种了非常多的树, 多到她都记不清了。但是她清晰地记得, 她种过的所有树都有两个共同点: 都是有根树, 且每个节点的子树个数均不超过 k 。

今天, 她发现森林中有一棵 n 个节点的树倒了, 已经无法分清哪个是原来的树根。因此, 她随便选取了一个节点为根, 记录了每个节点的父亲节点, 请你告诉她这棵树是否可能是她种的。如果可能, 请给出原来的根节点。

输入格式

第一行两个整数 n, k 。

第二行 n 个整数, 第 i 个整数为 p_i , 表示 i 号节点的父亲节点。特别地, 当前指定的根节点对应的 p_i 为 0。

输出格式

一行，一个整数。若可能是她种的树，输出任意一个可能的根节点；否则输出 -1 。

样例 #1

样例输入 #1

```
8 3
0 1 1 1 1 2 2 5
```

样例输出 #1

```
2
```

样例 #2

样例输入 #2

```
8 2
0 1 1 1 1 2 2 5
```

样例输出 #2

```
-1
```

样例 #3

样例输入 #3

```
1 14514
0
```

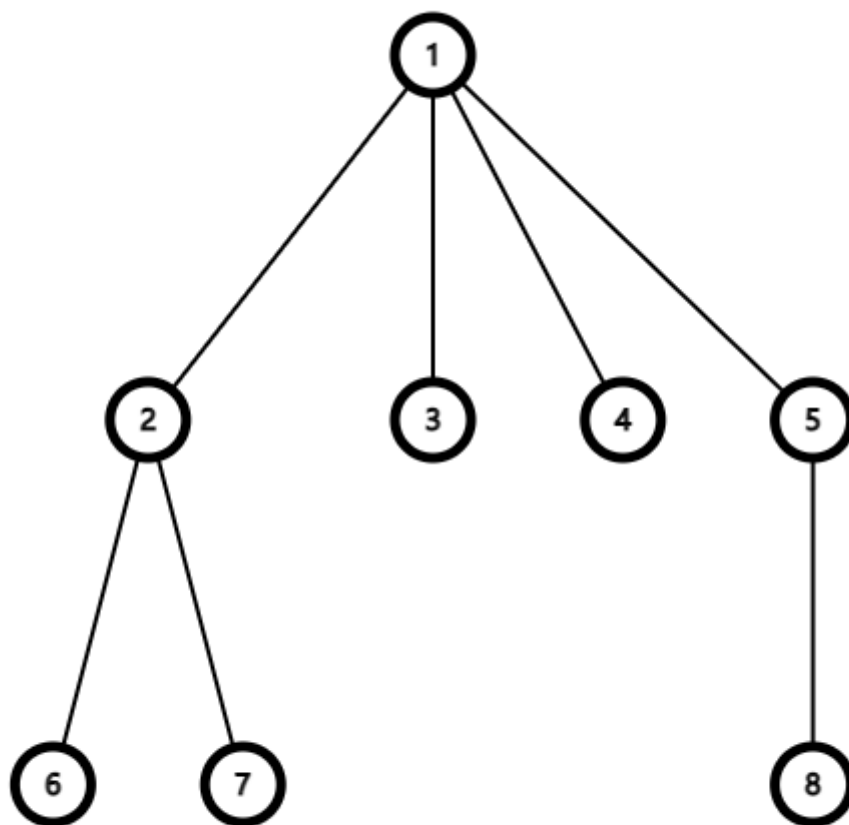
样例输出 #3

```
1
```

提示

样例 1 解释

样例如图所示：



洛谷@rui_er

虽然现在的形态不满足每个节点的子树个数不超过 3，但是如果以 2 号节点为根，就可以满足该要求。

样例 3 解释

k 只是每个节点的子树个数的上限，并不要求取到。

评分方式

本题采用自定义校验器 (Special Judge) 进行评分，一切符合要求的答案都将被认为是正确的。

数据范围

对于全部数据：

- $1 \leq n, k \leq 10^6$ 。
- 对于所有 $1 \leq i \leq n$, $0 \leq p_i \leq n$ 。

- 保证输入的 p 数组构成一棵树。

另有子任务如下：

- 子任务一 (20 分) : $n \leq 10^3$ 。
- 子任务二 (20 分) : $k \geq n$ 。
- 子任务三 (20 分) : $k = 1$ 。
- 子任务四 (40 分) : 无特殊性质。

「ZTOI R1」简单的题

题目描述

给定两个长度为 n 的序列 a, b , 求

$$\prod_{i=1}^n \sum_{j=1}^n \max_{k=1}^n \{ \min_{l=1}^n \{ a_i \times a_k, b_j \times b_l \} \}$$

由于答案可能会很大，所以请输出答案对 998244353 取模后的结果。

输入格式

第一行，一个正整数 n ，即数列长度。

第二行， n 个正整数，即数列 a 。

第三行， n 个正整数，即数列 b 。

输出格式

一行，一个正整数，即上述式子对 998244353 取模后的结果。

样例 #1

样例输入 #1

```
1
2
3
```

样例输出 #1

```
4
```

样例 #2

样例输入 #2

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

样例输出 #2

```
7630848
```

提示

样例 1 解释

原式 $= \min\{2 \times 2, 3 \times 3\} = \min\{4, 9\} = 4$ 。

数据范围

本题采用捆绑测试。

- 子任务一 (10 分) : $1 \leq n \leq 100$ 。
- 子任务二 (20 分) : $1 \leq n \leq 500$ 。
- 子任务三 (30 分) : $1 \leq n \leq 10^4$ 。
- 子任务四 (40 分) : 无特殊限制。

对于 100% 的数据, 有 $1 \leq n \leq 5 \times 10^5$, $1 \leq a_i \leq 10^9$ 。

「ZTOI R1」倒水问题

题目描述

你有 k 组水桶，每一组中有两个水桶，第 i 组的称为 A_i 和 B_i 。初始时，桶 A_i 的水量为 a_i ，桶 B_i 的水量为 $V - a_i$ 。

请你进行不超过 m 次操作，使得桶 A_i 的水量与目标水量 b_i 的绝对误差不超过 1。

一次操作如下：

- 你可以指定一个长度为 k 的数列 c ，满足 $c_i \in \{1, 2\}$ 。
- 对于每一个 $1 \leq i \leq k$ ，执行以下步骤：
 - 若 $c_i = 1$ ，从桶 B_i 中倒一半的水到桶 A_i 中。
 - 若 $c_i = 2$ ，从桶 A_i 中倒一半的水到桶 B_i 中。

输入格式

第一行三个整数 V, k, m 。

第二行 k 个浮点数代表 a_i 。

第三行 k 个浮点数代表 b_i 。

可以证明所有符合数据范围的输入均有解。所有浮点数均不超过两位小数。

输出格式

第一行为整数 n 代表操作次数，你需要保证 $0 \leq n \leq m$ 。

接下来 n 行，每行 k 个为 1 或为 2 的整数，表示你指定的数列 c 。

样例 #1

样例输入 #1

```
40 3 1
32.00 32.00 32.00
36.00 36.00 16.00
```

样例输出 #1

```
1
1 1 2
```

样例 #2**样例输入 #2**

```
100 3 10
0.56 49.12 47.89
3.83 67.98 39.01
```

样例输出 #2

```
10
1 2 1
1 2 1
1 2 1
2 1 1
2 1 2
1 1 2
2 2 2
2 1 1
2 2 1
2 1 2
```

提示**样例 1 解释**

初始时每个水桶的水量如下表：

	1	2	3
A	32	32	32
B	8	8	8

第一次操作将前两组的桶 B 倒一半水到桶 A ，将第三组的桶 A 倒一半水到桶 B ，新的水量如下表：

	1	2	3
A	36	36	16
B	4	4	24

符合目标水量的要求。

数据范围

测试点编号	k	V	m
1	$k \leq 1$	$V \leq 100$	$m = 10$
2	$k \leq 1$	$V \leq 10^3$	$m = 10$
3	$k \leq 1$	$V \leq 10^9$	$m = 32$
4	$k \leq 2$	$V \leq 10^3$	$m = 10$
5	$k \leq 2$	$V \leq 10^9$	$m = 32$
6	$k \leq 10^5$	$V \leq 100$	$m = 10$
7	$k \leq 10^5$	$V \leq 10^9$	$m = 100$
8	$k \leq 10^5$	$V \leq 10^9$	$m = 100$
9	$k \leq 10^5$	$V \leq 10^9$	$m = 32$
10	$k \leq 10^5$	$V \leq 10^9$	$m = 32$

对于 100% 的数据, $1 \leq a_i, b_i \leq k$ 。

「ZTOI R1」约数

题目描述

设 $\sigma(x) = \sum_{i=1}^x [i|x]$, 即 x 的因子个数, 其中 $[cond] = 1$ 当且仅当 $cond$ 为真。

设 $s(x) = \sum_{i=1}^x [i|x] \sigma(i)$ 。

现在给出 n , 求出 $\sum_{i=1}^n s(i)$ 。

输入格式

一个整数 n 。

输出格式

一个整数表示答案。

样例 #1

样例输入 #1

```
3
```

样例输出 #1

```
7
```

样例 #2

样例输入 #2

```
100
```

样例输出 #2

```
1471
```

提示

数据范围

- 25%: $1 \leq n \leq 1000$ 。
- 50%: $1 \leq n \leq 10^7$ 。
- 75%: $1 \leq n \leq 10^{10}$ 。

- 100%: $1 \leq n \leq 10^{12}$ 。

「ZTOI R1」千层蛋糕

题目背景

本题是一道 I/O 式交互题，但是交互只是用来强制在线的，所以推荐把本题当传统题做。

本题的时间限制为 800ms。

题目描述

小 R 是一个喜欢吃蛋糕的女孩子。她过生日的时候得到了一块超大的 n 层蛋糕，从下往上依次为第 $0 \sim n-1$ 层。蛋糕有许多不同的颜色，对应不同的口味，第 i 层蛋糕的颜色被记为正整数 a_i 。

她定义蛋糕第 $l \sim r$ 层的可爱度为这段区间的颜色段数量。例如，如果第 $l \sim r$ 层的颜色为

`[\color{#e64980}1,1\color{black},\color{#be4bdb}2\color{black},\color{#e64980}1\color{black},\color{#228be6}3,3,3\color{black}]`，则可爱度为 4。

另外她还会时不时地对蛋糕的每一层的顺序进行重排。具体地，如果这次重排的参数为 x ，她会把所有满足 $i \oplus j = x$ 的下标 i, j 对应的两层蛋糕交换。

为方便处理，两种操作的代号如下：

- **1 x**：以 x 为重排的参数对蛋糕进行重排。
- **2 l r**：查询第 $l \sim r$ 层蛋糕的可爱度。

请你帮忙实现这两种操作。

其中 \oplus 表示按位异或运算，即 C++ 语言的 `^` 或 `xor`。

交互方式

本题为 I/O 式交互题。除子任务四外，本题强制在线。

在程序开始运行时，可以首先读入的信息如下：

- 第一行三个整数 T, k, m ，其中 $n = 2^k$ 。
- 第二行 n 个整数，表示蛋糕每一层的颜色 a_i ，按下标顺序依次给出。

如果 $T = 0$ ：

- 你可以继续读入如下信息：
 - 接下来 m 行，每行两或三个整数，描述一次操作。
- 你可以随时输出答案。

如果 $T=1$:

- 你可以继续读入如下信息共 m 次：
 - 一行，两或三个整数，描述一次操作。
- 你必须在得到操作二后立即输出答案，否则无法继续读入。

在每次输出答案后，你需要清空缓冲区。你可以使用如下语句来清空缓冲区：

- 对于 C/C++ 语言：`fflush(stdout)`。
- 对于 C++ 语言：`std::cout << std::flush`。

特别地，对于 C++ 语言，如果使用 `std::endl` 而不是 `'\n'` 输出换行，也可以清空缓冲区。

如果你第一次接触 I/O 式交互题，请参考下方「提示说明」中的样例程序。

输入格式

以下为样例中的输入格式。

第一行三个整数 T, k, m ，其中 $n=2^k$ 。

第二行 n 个整数，表示蛋糕每一层的颜色 a_i ，按下标顺序依次给出。

接下来 m 行，每行两或三个整数，描述一次操作。

输出格式

以下为样例中的输出格式。

若干行，对于每个操作 2 回答询问。

样例 #1

样例输入 #1

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

样例输出 #1

```
5
4
```

样例 #2

样例输入 #2

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

样例输出 #2

```
5
4
```

提示

样例 1 解释

此数据点允许离线。

初始时蛋糕每一层的颜色为 $[1, 2, 1, 3, 2, 4, 5, 1]$ 。

第 $1 \sim 5$ 层的颜色为 $[2, 1, 3, 2, 4]$ ，它的可爱度为 5。

进行重排操作后，蛋糕每一层的颜色变为 $[3, 1, 2, 1, 1, 5, 4, 2]$ 。

第 $1 \sim 5$ 层的颜色为 $[1, 2, 1, 1, 5]$ ，它的可爱度为 4。

样例 2 解释

此数据点强制在线。

样例程序

本题的样例程序（仅为交互方式的示例）。

数据范围

对于全部数据：

- $T \in \{0, 1\}$ 。

- $0 \leq k \leq 17, n=2^k$ 。
- $1 \leq m \leq 7 \times 10^4$ 。
- 对于所有 $0 \leq i < n, 1 \leq a_i \leq 10^9$ 。
- 对于所有操作一, $0 \leq x < n$ 。
- 对于所有操作二, $0 \leq l \leq r < n$ 。

另有子任务如下：

- 子任务一 (15 分) : $T=1, k \leq 10, m \leq 10^3$ 。
- 子任务二 (15 分) : $T=1$, 不存在操作一。
- 子任务三 (20 分) : $T=1$, 对于所有操作二, $l=0, r=n-1$ 。
- 子任务四 (20 分) : $T=0$ 。
- 子任务五 (30 分) : $T=1$ 。

为什么不使用更常用的异或上次答案强制在线？

本来是这种强制在线方式，但是这样的话子任务三就非常方便做了：每次读入的 l 即为上次操作的答案，最后一次操作暴力。

因此我们不得已采用 I/O 交互的方式强制在线。

「ZTOI R1」凸问题

题目描述

有一个 n 个点的凸多边形，按照顺时针或者逆时针方向给出

你需要选择 k 个点，使得这 k 个点形成的凸包的周长最长。

你需要输出一组方案

输入格式

第一行两个整数 n, k

接下来 n 行，每行两个不超过4位的实数代表凸包上的一个点。

输出格式

第一行一个实数代表答案。

第二行输出你选择的点的个数。

第三行 k 个数，**从小到大**，代表你选择的点的编号。

答案与标准答案的绝对误差或者相对误差小于等于 10^{-6} 被认为是正确的。

样例 #1

样例输入 #1

```
8 3
1.00 2.00
2.00 1.00
2.00 -1.00
1.00 -2.00
-1.00 -2.00
-2.00 -1.00
-2.00 1.00
-1.00 2.00
```

样例输出 #1

```
11.404918347
3
1 3 6
```

提示

对于 100% 的数据, $2 \leq k \leq n \leq 2000$

点的坐标在 $[-10^9, 10^9]$ 之内

数据有梯度

「ZTOI R1」字符串问题

题目描述

给定两个长度为 n, m 的字符串 s, t 。记字符串 a 的下标在区间 $[l, r]$ 内的子串为 $a_{l \dots r}$ 。

你需要回答 Q 次询问, 每次询问给出 l_1, r_1, l_2, r_2 , 请计算通过下面三种操作将 $s_{l_1 \dots r_1}$ 变为 $t_{l_2 \dots r_2}$ 的最小代价:

1. 在 s 中添加一个字符 x 。此操作需要花费 A_x 的代价。
2. 在 s 中删除一个字符 x 。此操作需要花费 B_x 的代价。

3. 在 s 中将一个字符 x 改为字符 y 。此操作需要花费 $C_{x,y}$ 的代价。

特别地，如果 A, B, C 中有元素为 -1 ，代表不能进行这个操作。

询问之间相互独立。

输入格式

第 $1 \sim 26$ 行，一个大小为 26×26 的整数矩阵，第 i 行第 j 列的元素为 $C_{i,j}$ 。

第 27 行，一个长度为 26 的整数数列，第 i 项为 A_i 。

第 28 行，一个长度为 26 的整数数列，第 i 项为 B_i 。

第 29 行，三个整数 n, m, q 。

第 30 行，长度为 n 的字符串 s 。

第 31 行，长度为 m 的字符串 t 。

第 $32 \sim 32 + q - 1$ 行，每行四个整数 l_1, r_1, l_2, r_2 。

输出格式

共 q 行，每行一个整数代表询问的答案。特别地，若无法达成目标，输出 -1 。

样例 #1

样例输入 #1

```
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
```

```

1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1
1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1 1

```

20 20 20

acaaabccabccabab

bbcabbccbaaaccaaccba

3 9 3 17

1 1 3 4

5 19 11 16

1 3 5 13

3 6 9 13

1 11 5 17

1 18 1 14

5 12 11 19

5 11 11 13

5 13 1 1

1 15 1 6

5 20 1 7

6 13 5 11

1 7 1 13

1 19 16 17

1 13 10 17

3 9 7 17

1 11 11 13

1 20 2 19

1 17 5 11

样例输出 #1

```
10
1
9
7
2
8
9
5
5
8
9
9
2
9
17
6
7
8
9
10
```

提示

样例解释

即求 $s_{l_1 \dots r_1}$ 和 $t_{l_2 \dots r_2}$ 的编辑距离。

数据范围

- 子任务一 (30 分) : $1 \leq n, m, q \leq 100$ 。
- 子任务二 (20 分) : $l_2 = r_2$ 。
- 子任务三 (20 分) : $n = 2$ 。
- 子任务四 (30 分) : 无特殊性质。

对于 100% 的数据, $1 \leq n \times m \leq 10^5$, $n, m \geq 1$, $1 \leq q \leq 10^5$, $1 \leq A_i, B_i, C_{i,j} \leq 10^9$, $1 \leq l_1 \leq r_1 \leq n$, $1 \leq l_2 \leq r_2 \leq m$ 。