

NOIP 模拟赛

Gold14526 & szt100309

08:00~11:40

题目名称	镜之寺庙	沉思	山顶	核心
题目类型	传统型	传统型	传统型	传统型
目录	temple	reflection	summit	core
文件名	temple	reflection	summit	core
输入文件名	temple.in	reflection.in	summit.in	core.in
输出文件名	temple.out	reflection.out	summit.out	core.out
时间限制	4.0 秒	3.0 秒	4.0 秒	2.0 秒
内存限制	64~2048 MB	512 MB	512 MB	1024 MB
测试包数目	6	8	6	13
测试包等分	否	否	否	否

提交源程序

C++ 语言	temple.cpp	reflection.cpp	summit.cpp	core.cpp
--------	------------	----------------	------------	----------

编译选项

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

注意事项（请仔细阅读）

1. 文件名（程序名和输入输出文件名）必须使用英文小写。
2. 赛时请勿大声嘴巴。
3. 若无特殊说明，结果的比较方式为全文比较（过滤行末空格及文末回车）。
4. 选手提交的程序源文件必须不大于 100KB。
5. 这是一场你真正能获得 404 分的模拟赛。赛时，你每在一道题中获得 101 分，你都将获得一瓶海之言，**每道题的最后一个子任务都依赖这道题的其它所有子任务**。

镜之寺庙 (temple)

【题目背景】



如果这些都是存在于我脑海中，为什么它们会对我充满敌意？

这些事情一直在骚扰我。

我是我自己世界的入侵者。

呼吸，这些都来自我的内心。

站起来，Madeline。

想想羽毛。

你能拯救 Theo。

【问题描述】

给定一棵 n 个点的树，第 i 个点有权值 a_i 。

q 次询问，每次询问给出两个数 u, v ，表示查询 u 到 v 路径上是否有两个节点的权值相同。

输入经过加密，开始时给定 $base, mod$ ，每次询问实际输入的数为 $u' = u \oplus hashans, v' = v \oplus hashans$ ，设第 i 次询问的答案为 ans_i ，那么第 i 次询问的 $hashans$ 为 $(\sum_{p=1}^{i-1} [ans_p = "Yes"] \times base^{i-p-1}) \bmod mod$ 。

【输入格式】

从文件 **temple.in** 中读入数据。

第一行输入三个数 $n, base, mod$ 。

第二行 n 个数，代表序列 a 。

接下来 $n - 1$ 行，每行两个数 x, y ，代表 x, y 之间有一条边。

接下来一行一个数 q 。

接下来 q 行，每行两个数 u', v' ，代表加密后的一次询问。

【输出格式】

输出到文件 *temple.out* 中。

对于每个询问，输出一行“Yes”或“No”，表示答案。

【样例 1 输入】

```
1 7 0 1
2 3 1 4 1 5 3 2
3 1 3
4 2 3
5 4 3
6 5 1
7 6 2
8 7 5
9 5
10 6 4
11 2 4
12 1 5
13 7 2
14 1 1
```

【样例 1 输出】

```
1 Yes
2 Yes
3 No
4 No
5 No
```

【样例 2 输入】

```
1 12 0 1
2 5 8 2 10 5 1 12 8 2 7 11 4
3 1 4
4 2 4
5 3 4
6 5 1
7 6 1
8 7 2
9 8 2
10 9 3
11 10 3
12 11 7
13 12 10
14 6
15 5 9
16 11 8
17 6 12
18 1 11
19 9 3
20 10 10
```

【样例 2 输出】

```
1 Yes
2 Yes
3 No
4 No
5 Yes
6 No
```

【样例 3】

见选手目录下的 *temple/temple3.in* 与 *temple/temple3.ans*。
该样例符合子任务 1 的约束。

【样例 4】

见选手目录下的 *temple/temple4.in* 与 *temple/temple4.ans*。
该样例符合子任务 2 的约束。

【样例 5】

见选手目录下的 *temple/temple5.in* 与 *temple/temple5.ans*。
该样例符合子任务 3 的约束。

【样例 6】

见选手目录下的 *temple/temple6.in* 与 *temple/temple6.ans*。
该样例符合子任务 4 的约束。

【样例 7】

见选手目录下的 *temple/temple7.in* 与 *temple/temple7.ans*。
该样例符合子任务 5 的约束。

【测试点约束】

对于所有数据，保证：

- $1 \leq n \leq 10^5$;
- $0 \leq base < mod \leq 10^5 + 3$;
- $1 \leq a_i \leq n$;
- $1 \leq q \leq 10^5$;
- $1 \leq x, y \leq n$;
- $1 \leq u, v \leq n$;
- 保证给出的图是一棵树

本题采用捆绑测试，各子任务特殊限制如下：

子任务	$n \leq$	$q \leq$	$mod =$	特殊性质	空间限制	分值	
1	5000	5000	1	无	2048MB	12	
2	10^5	10^5		A		18	
3				B		20	
4				C		24	
5				无		26	
6	10^5	10^5	$10^5 + 3$	无	64MB	1	

特殊性质 A: 树是一条链。

特殊性质 B: 树等概率从所有可能的树中随机生成一种。

特殊性质 C: 树是一条毛毛虫。

沉思 (reflection)

【题目背景】



“我帮过你的忙。”
“你算不上是登山者！”
“我只是在努力帮你！”

【问题描述】

给定长度为 n 的序列 a 。

我们定义 $f(l, r, x)$:

$$f(l, r, x) = \begin{cases} |a_l - x| & l = r \\ |a_r - f(l, r - 1, x)| & l < r \end{cases}$$

q 次询问，每次给定三个数 l, r, x ，查询 $f(l, r, x)$ 的值。

输入经过加密，开始时给定 c ，每次询问实际输入的数为 $l' = l \oplus (c \times lstans), r' = r \oplus (c \times lstans), x' = x \oplus (c \times lstans)$ ，其中 $lstans$ 为上次询问的答案。特别的，若当前是第一次询问，则 $lstans = 0$ 。

【输入格式】

从文件 **reflection.in** 中读入数据。

第一行输入三个数 n, q, c 。

第二行输入 n 个数，代表序列 a 。

接下来 q 行，每行三个数 l', r', x' ，代表加密后的一次询问。

【输出格式】

输出到文件 *reflection.out* 中。

对于每次询问，输出一行一个数，代表答案。

【样例 1 输入】

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

【样例 1 输出】

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

【样例 2 输入】

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

【样例 2 输出】

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

【样例 3】

见选手目录下的 *reflection/reflection3.in* 与 *reflection/reflection3.ans*。
该样例符合子任务 1 的约束。

【样例 4】

见选手目录下的 *reflection/reflection4.in* 与 *reflection/reflection4.ans*。
该样例符合子任务 2 的约束。

【样例 5】

见选手目录下的 *reflection/reflection5.in* 与 *reflection/reflection5.ans*。
该样例符合子任务 3 的约束。

【样例 6】

见选手目录下的 *reflection/reflection6.in* 与 *reflection/reflection6.ans*。
该样例符合子任务 4 的约束。

【样例 7】

见选手目录下的 *reflection/reflection7.in* 与 *reflection/reflection7.ans*。
该样例符合子任务 5 的约束。

【样例 8】

见选手目录下的 *reflection/reflection8.in* 与 *reflection/reflection8.ans*。
该样例符合子任务 6 的约束。

【样例 9】

见选手目录下的 *reflection/reflection9.in* 与 *reflection/reflection9.ans*。
该样例符合子任务 7 的约束。

【测试点约束】

以下设单个测试点内的 x 和 a_i 的最大值为 V 。

对于所有数据，保证：

- $1 \leq n \leq 2 \times 10^5$;
- $1 \leq q \leq 2 \times 10^5$;
- $c \in \{0, 1\}$;
- $1 \leq l \leq r \leq n, 1 \leq x \leq 2 \times 10^5$
- $\forall 1 \leq i \leq n, 1 \leq a_i \leq 2 \times 10^5$;

本题采用捆绑测试，各子任务特殊限制如下：

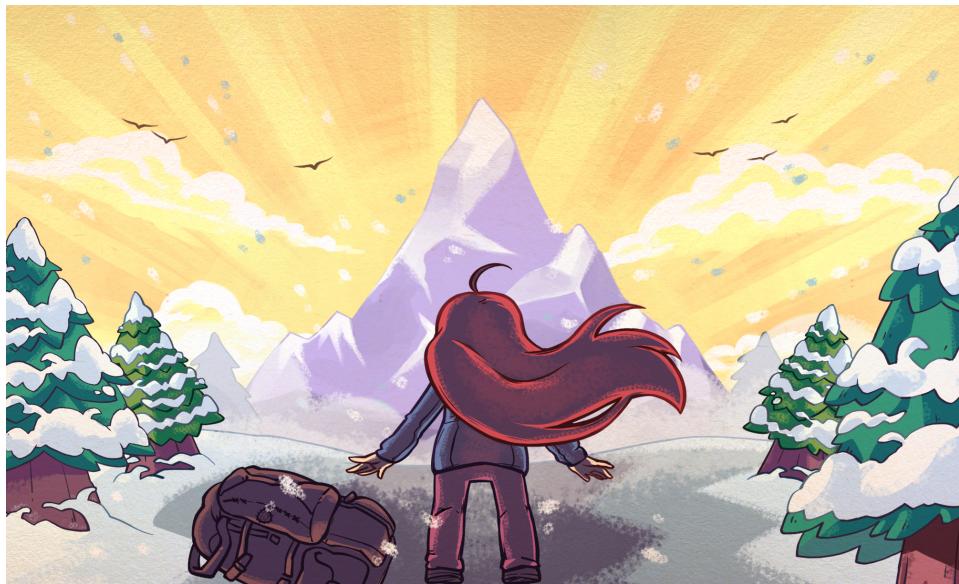
子任务	$n \leq$	$q \leq$	$V \leq$	$c =$	特殊性质	分值	
1	10^4	10^4	2×10^5	0	无	8	
2	10^5	10^5	100		A	9	
3					B	9	
4					无	9	
5	2×10^5	2×10^5	2×10^5		AB	18	
6					A	20	
7					无	27	
8					无	1	

特殊性质 A: $l = 1$ 。

特殊性质 B: $r = n$ 。

山顶 (summit)

【题目背景】



“我们真的做到了。”
“我简直不敢相信！”
“哇噢！太漂亮了！”
“看看我们走了多远了！”
“我真高兴我能先看到这个，然后才……”
“嗯，然后我才……”
“噢，好吧。”
“一旦我们走下那座山，你就不能再跟我说话了……”
“至少不是像这样说话。”
“是的。”
“我知道我们最后还是要回家的。”
“但是舍弃这具躯壳还是很让人害怕。”
“但你还是会存在。”
“我们来这里以前，你就是另一个我。”
“是的，我知道。”
“但是，我会依靠你来倾听我的想法。”
“……哪怕我再也不能自己跟你说话了。”
“我只是……不能回到过去那种样子了。”
“来这山之前的样子。”

“别担心，我也不可能回到以前那个样子。”
“我真希望我们从今往后能齐心协力。”
“我相信你。”
“可我还是紧张……”
“那么……现在要怎么做？”
“……”
“就让我们沉浸于当下吧。”

【问题描述】

给定长度为 n 的正整数序列 a, b , 你需要选择长度为 k 的两个正整数序列 p, q , 使得其中所有数都在 $[1, n]$ 中, 并且:

- $\forall 1 \leq i < k, p_i < p_{i+1}, q_i < q_{i+1}$;
- $\forall 1 \leq i \leq k, p_i \leq q_i$.

你需要让 $\sum_{i=1}^k a_{p_i} + \sum_{i=1}^k b_{q_i}$ 最小。

对于 $k = l, l+1, \dots, r$ 都输出这个最小值。

【输入格式】

从文件 *summit.in* 中读入数据。

本题有多组测试数据, 第一行输入一个正整数 T , 代表数据组数。

对于每组数据:

第一行输入三个数 n, l, r 。

第二行输入 n 个数, 代表序列 a 。

第三行输入 n 个数, 代表序列 b 。

【输出格式】

输出到文件 *summit.out* 中。

对于每组数据, 输出一行 $r - l + 1$ 个数, 分别代表 $k = l, l+1, \dots, r$ 时的答案。

【样例 1 输入】

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

```
5 8 4 4
6 3 8 7 9 9 4 6 8
7 2 5 9 4 3 8 9 1
8 6 1 1
9 1 2 3 4 5 6
10 6 5 4 3 2 1
11 6 1 1
12 6 5 4 3 2 1
13 1 2 3 4 5 6
```

【样例 1 输出】

```
1 17
2 32
3 2
4 7
```

【样例 2 ~ 3】

见选手目录下的 *summit/summit2~3.in* 与 *summit/summit2~3.ans*。
这些样例符合子任务 1 的约束。

【样例 4 ~ 5】

见选手目录下的 *summit/summit4~5.in* 与 *summit/summit4~5.ans*。
这些样例符合子任务 2 的约束。

【样例 6 ~ 7】

见选手目录下的 *summit/summit6~7.in* 与 *summit/summit6~7.ans*。
这些样例符合子任务 3 的约束。

【样例 8 ~ 9】

见选手目录下的 *summit/summit8~9.in* 与 *summit/summit8~9.ans*。
这些样例符合子任务 4 的约束。

【样例 10 ~ 11】

见选手目录下的 *summit/summit10~11.in* 与 *summit/summit10~11.ans*。
这些样例符合子任务 5 的约束。

【测试点约束】

记 $\sum n, \sum r$ 分别表示单个测试点内所有 n, r 的和。

对于所有数据，保证：

- $1 \leq T \leq 10^4$;
- $1 \leq n, \sum n \leq 2 \times 10^6$;
- $1 \leq l \leq r \leq n$;
- $1 \leq \sum r \leq 5 \times 10^5$
- $\forall 1 \leq i \leq n, 1 \leq a_i, b_i \leq 10^9$;

本题采用捆绑测试，各子任务特殊限制如下：

子任务	$\sum n \leq$	$\sum r \leq$	$l =$	分值
1	10	10	r	8
2	20	20		11
3	400	400		25
4	2000	2000		28
5	3×10^5	3×10^5		28
6	2×10^6	5×10^5	1	1

核心 (core)

【题目背景】



.....就这样?
这一切的核心就是这个?
多么空洞.....深广.....
感觉又让人有些怀念.....
让人觉得平静。

【问题描述】

给定 n, m, k , 和由括号组成的字符串 $S_{1 \sim n}, T_{1 \sim m}$ 。

定义 $f(i, j)$ 为: 以 S_i 为前缀, 以 T_j 为后缀的长度为 k 的匹配括号串数量。

求 $\sum_{i=1}^n \sum_{j=1}^m f(i, j)$ 。

【输入格式】

从文件 **core.in** 中读入数据。

第一行输入一个数 id , 表示测试点所在的子任务编号, 特别的, 对于样例, $id = 0$ 。

第二行输入三个数 n, m, k 。

接下来 n 行, 每行一个括号串, 代表 $S_{1 \sim n}$ 。

接下来 m 行, 每行一个括号串, 代表 $T_{1 \sim m}$ 。

【输出格式】

输出到文件 *core.out* 中。
输出一行一个数代表答案。

【样例 1 输入】

```
1 0
2 1 2 6
3 (
4 ())
5 ()()
```

【样例 1 输出】

```
1 4
```

【样例 2 输入】

```
1 0
2 2 2 4
3 ((((
4 ()()
5 )())
6 ())
```

【样例 2 输出】

```
1 2
```

【样例 3】

见选手目录下的 *core/core3.in* 与 *core/core3.ans*。
该样例符合子任务 1 的约束。

【样例 4】

见选手目录下的 *core/core4.in* 与 *core/core4.ans*。
该样例符合子任务 2 的约束。

【样例 5】

见选手目录下的 *core/core5.in* 与 *core/core5.ans*。
该样例符合子任务 3 的约束。

【样例 6】

见选手目录下的 *core/core6.in* 与 *core/core6.ans*。
该样例符合子任务 4 的约束。

【样例 7】

见选手目录下的 *core/core7.in* 与 *core/core7.ans*。
该样例符合子任务 5 的约束。

【样例 8】

见选手目录下的 *core/core8.in* 与 *core/core8.ans*。
该样例符合子任务 6 的约束。

【样例 9】

见选手目录下的 *core/core9.in* 与 *core/core9.ans*。
该样例符合子任务 7 的约束。

【样例 10】

见选手目录下的 *core/core10.in* 与 *core/core10.ans*。
该样例符合子任务 8 的约束。

【样例 11】

见选手目录下的 *core/core11.in* 与 *core/core11.ans*。
该样例符合子任务 9 的约束。

【样例 12】

见选手目录下的 *core/core12.in* 与 *core/core12.ans*。
该样例符合子任务 10 的约束。

【样例 13】

见选手目录下的 *core/core13.in* 与 *core/core13.ans*。
该样例符合子任务 11 的约束。

【样例 14】

见选手目录下的 *core/core14.in* 与 *core/core14.ans*。
该样例符合子任务 12 的约束。

【测试点约束】

以下记 L 表示所有括号串的长度之和。

对于所有数据，保证：

- $1 \leq n \leq 10^6$;
- $1 \leq m \leq 10^6$;
- $1 \leq k \leq 10^7$;
- k 为偶数;
- $\forall 1 \leq i \leq n, 1 \leq |S_i| \leq 10^7$;
- $\forall 1 \leq i \leq m, 1 \leq |T_i| \leq 10^7$;
- $1 \leq L \leq 10^7$ 。

本题采用捆绑测试，各子任务特殊限制如下：

子任务	$n, m \leq$	$L \leq$	$k \leq$	特殊性质	分值
1	10	100	15		8
2	50	5000	100		8
3	250		5000	无	8
4	5000				9
5	5×10^4	5×10^5	2×10^5	A	6
6				B	6
7				C	9
8				无	8
9	2×10^5	10^6		A	8
10				B	8
11				C	10
12				无	12
13	10^6	10^7	10^7	无	1

特殊性质 A: $\forall 1 \leq i \leq n \wedge 1 \leq j \leq m, |S_i| + |T_j| \geq k$ 。

特殊性质 B: 保证每个括号串的左右括号数量都相同。

特殊性质 C: $\forall 1 \leq i \leq n \wedge 1 \leq j \leq m, |S_i| + |T_j| \leq k$ 。