

Clariss' Contest # 1

By Claris

2016 年 4 月 30 日

题目名称	二进制的世界	可持久化字符串	反函数
输入文件名	binary.in	string.in	inverse.in
输出文件名	binary.out	string.out	inverse.out
每个测试点时限	1s	1s	1s
测试点数目	10	10	10
每个测试点分值	10	10	10
内存限制	512MB	512MB	512MB
是否有部分分	否	否	否
题目类型	传统	传统	传统

注意：第三题评测时将开启-O2优化开关。

1 二进制的世界(binary.c/cpp/pas)

1.1 题目描述

在二进制的世界里，只有三种运算：按位与(and)、按位或(or)、按位异或(xor)。

在这个世界里，有 n 个人从左往右排成一排，其中从左往右第 i 个人手上写着一个数字 a_i 。

每个人都想在他左边找到他的好朋友，第 i 个人和第 j 个人之间的友好程度为 $a_i \text{ opt } a_j$ ，其中 opt 是三种运算中的一种。

请对于从左往右第2到第 n 个人，求出他左边和他友好程度最大的人，以及达到最大值的人数。

1.2 输入格式

第一行包含一个正整数 n ，一个串 opt 和一个整数 type ，分别表示人数，运算类型以及数据类型。

第二行包含 n 个整数，其中第 i 个数为 a_i ，表示从左往右第 i 个人手上的数字。

1.3 输出格式

如果 $\text{type} = 0$ ，输出 $n - 1$ 行，每行一个整数，其中第 i 行输出和第 $i + 1$ 个人的友好程度的最大值。

如果 $\text{type} = 1$ ，输出 $n - 1$ 行，每行两个整数，其中第 i 行先输出和第 $i + 1$ 个人的友好程度的最大值，再输出达到最大值的人数。

1.4 样例输入

```
5 and 1
3 5 2 7 1
```

1.5 样例输出

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

1.6 数据范围与约定

对于100%的数据， $0 \leq a_i \leq 65535$ 。

测试点编号	n	opt	type
1	= 1000	and	1
2	= 1000	or	1
3	= 100000	xor	1
4	= 100000	xor	1
5	= 100000	and	0
6	= 100000	or	0
7	= 100000	and	1
8	= 100000	and	1
9	= 100000	or	1
10	= 100000	or	1

2 可持久化字符串(string.c/cpp/pas)

2.1 题目描述

一个串 T 是 S 的循环节，当且仅当存在正整数 k ，使得 S 是 T^k (即 T 重复 k 次)的前缀，比如abcd是abcdabcdab的循环节。

维护一个字符串 S ，一开始是空串，进行 m 次操作，每次操作包含两个整数 x_i, c_i ，表示这次操作的字符串为在第 x_i 次操作之后的字符串末尾添加一个字符 c_i 所形成的字符串。

请在每次操作完毕之后，求出该次操作得到的字符串最短的循环节的长度。

2.2 输入格式

第一行包含三个整数 $n, m, type$ ，分别表示操作总数，字符集大小以及数据类型。

接下来 n 行，第 i 行包含两个整数 x_i, c_i ，表示一次操作，若 $x_i = 0$ 则表示初始串，即空串。

如果 $type = 1$ ，说明数据进行了加密， x 和 c 都需要异或上一次的输出，如果是第一次操作，那么视上一次的输出为0。

2.3 输出格式

输出 n 行，每行一个正整数，表示这次操作得到的字符串最短的循环节的长度。

2.4 样例输入

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

2.5 样例输出

```
1
2
2
2
1
```

2.6 数据范围与约定

对于100%的数据, $0 \leq x_i < i$, $1 \leq c_i \leq m$ 。

测试点编号	n	m	$type$	其它约定
1	= 100	= 100	1	无
2	= 100	= 100	1	无
3	= 300000	= 300000	1	$x_i = i - 1$
4	= 300000	= 300000	1	$x_i = i - 1$
5	= 300000	= 10	0	无
6	= 300000	= 10	0	无
7	= 300000	= 10	0	无
8	= 300000	= 300000	1	无
9	= 300000	= 300000	1	无
10	= 300000	= 300000	1	无

3 反函数(inverse.c/cpp/pas)

3.1 题目描述

一个括号串是平衡的当且仅当它是下面几种情况中的一种：

1. 括号串()是平衡的。
2. 若A,B均为平衡的括号串，则串AB也是平衡的。
3. 若A为平衡的括号串，则串(A)也是平衡的。

给定一棵有 n 个节点的无根树，每个节点要么是(，要么是)。

定义 $S(x, y)$ 为从 x 开始沿着最短路走到 y ，将沿途经过的点上的字符依次连起来得到的字符串。

函数 $f(x, y)$ 等于对 $S(x, y)$ 进行划分，使得每一个部分都是平衡的括号串，能得到的最大的段数。比如 $((()))()$ 的最大段数为3， $((()))()$ 的最大段数为2。

特别地，如果 $S(x, y)$ 本身就不平衡，那么 $f(x, y) = 0$ 。

对于给定的 x, y ， $f(x, y)$ 的求解是非常容易的，但是输入 k ，统计有多少 x, y 满足 $1 \leq x, y \leq n$ ，且 $f(x, y) = k$ 却非常棘手。你能写一个程序解决这个问题吗？

3.2 输入格式

第一行包含一个正整数 n ，表示节点数。

接下来 $n - 1$ 行，每行包含两个正整数 x, y ，表示 x 和 y 之间有一条边。

接下来 n 行，每行一个字符(或)，其中第 i 行表示 i 号点的字符。

接下来一行包含一个正整数 m ，表示询问的个数。

接下来 m 行，每行一个正整数 k ，表示一个询问。

3.3 输出格式

输出 m 行，每行一个整数，表示有多少 x, y 满足 $1 \leq x, y \leq n$ ，且 $f(x, y) = k$ 。

3.4 样例输入

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

3.5 样例输出

4

2

0

3.6 数据范围与约定

对于100%的数据， $1 \leq x, y \leq n$ ， $1 \leq k \leq \frac{n}{2}$ 。

测试点编号	n	m
1	= 100	= 50
2	= 5000	= 2500
3	= 5000	= 2500
4	= 50000	= 10
5	= 50000	= 10
6	= 50000	= 10
7	= 50000	= 25000
8	= 50000	= 25000
9	= 50000	= 25000
10	= 50000	= 25000