

ION 1202 模拟赛

ASDFZ

时间：1202 年 114 月 514 日 08:00 ~ 13:00

题目名称	消除	排序	括号
题目类型	传统型	传统型	传统型
目录	increase	shuffle	bad
可执行文件名	increase	shuffle	bad
输入文件名	increase.in	shuffle.in	bad.in
输出文件名	increase.out	shuffle.out	bad.out
每个测试点时限	1.0 秒	1.0 秒	2.0 秒
内存限制	64 MiB	256 MiB	1 GiB
子任务数目	4	4	4
测试点是否等分	否	否	否

提交源程序文件名

对于 C++ 语言	increase.cpp	shuffle.cpp	bad.cpp
-----------	--------------	-------------	---------

编译选项

对于 C++ 语言	-lm -O2 -std=c++11
-----------	--------------------

注意事项与提醒：

1. 提交时需要建子文件夹。
2. 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）。
3. 程序可使用的栈内存空间限制与题目的内存限制一致。
4. 评测机很快，所有题的时间限制至少为标程用时的两倍。

消除 (increase)

题目描述

请注意本题的时空限制。

给定一棵以 1 为根的树。每次随机选择树上未被删去的一个点，将它和它的子树中所有点删去。如果选到根，就把整棵树删去，并停止这一过程。求 k 次内能把这棵树删空的概率，答案对 998244353 取模。

输入格式

第一行两个正整数 n 和 k ，意义如题面所示。

接下来 $n - 1$ 行，每行两个正整数 $u, v (1 \leq u, v \leq n)$ ，描述了树上一条边 (u, v) 。

输出格式

一行一个整数，表示答案对 998244353 取模的结果。

样例 1 输入

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

样例 1 输出

```
124780545
```

样例 2 输入

详见选手文件夹下的 `increase/increase2.in` 文件。
这组样例满足测试点 1 的约束。

样例 2 输出

详见选手文件夹下的 `increase/increase2.ans` 文件。

样例 3 输入

详见选手文件夹下的 `increase/increase3.in` 文件。
这组样例满足测试点 3 的约束。

样例 3 输出

详见选手文件夹下的 `increase/increase3.ans` 文件。

数据范围与约束

本题采用捆绑测试，共 4 个测试点，你必须通过每个测试点中的所有数据才能获得该测试点的分数。

对于 100% 的数据，满足： $1 \leq k \leq n \leq 500$ 。

具体的数据范围见下表：

测试点编号	n	特殊性质	分数
1	≤ 150	对于第 i 条边，满足 $u = i, v = i + 1$	10
2	≤ 50	无	40
3	≤ 150	无	30
4	≤ 500	无	20

排序 (shuffle)

题目描述

给定一个长度为 n 的 01 串 S ，可以进行两种操作：

1. 将 S 中的最后一个字符删去，放入序列开头。
2. 将 S 中的最后一个字符删去，放入第二个数前。

求使该串升序排列的最小操作次数。

输入格式

第一行一个正整数 n ，表示 S 的长度。

第二行一个 01 串 S 。

输出格式

一行一个整数，表示最少操作次数。

样例 1 输入

```
4
1010
```

样例 1 输出

```
3
```

样例 1 解释

$1010 \rightarrow 0101 \rightarrow 0110 \rightarrow 0011$ 。

样例 2 输入

```
6
001001
```

样例 2 输出

```
7
```

样例 2 解释

001001 \rightarrow 100100 \rightarrow 100010 \rightarrow 010001 \rightarrow 011000 \rightarrow 001100 \rightarrow 000110 \rightarrow 000011

样例 3 输入

```
8
01010101
```

样例 3 输出

```
11
```

样例 4 输入

详见选手文件夹下的 `shuffle/shuffle4.in` 文件。
这组样例满足测试点 3 的约束。

样例 4 输出

详见选手文件夹下的 `shuffle/shuffle4.ans` 文件。

样例 5 输入

详见选手文件夹下的 `shuffle/shuffle5.in` 文件。
这组样例满足测试点 4 的约束。

样例 5 输出

详见选手文件夹下的 `shuffle/shuffle5.ans` 文件。

数据范围与约束

本题采用捆绑测试，共 4 个测试点，你必须通过每个测试点中的所有数据才能获得该测试点的分数。

对于 100% 的数据，满足： $1 \leq n \leq 10^5$ 。

具体的数据范围见下表：

测试点编号	n	分数
1	≤ 20	10
2	≤ 100	30
3	$\leq 10^3$	20
4	$\leq 10^5$	40

括号 (bad)

题目描述

给定一个长度为 n 的括号序列 c ，定义好的括号序列如下：

- 空串是好的括号序列。
- 若 A 和 B 是好的括号序列，则 AB 也是好的括号序列。
- 若 A 是好的括号序列，则 (A) 也是好的括号序列。

定义一个 $1, 2, \dots, n$ 的排列 p 是好的，当且仅当括号序列 $d(\forall i \in [1, n], d_i = c_{p_i})$ 是合法的括号序列，定义 c 的价值为所有好的排列的逆序数总和。

c 并不稳定，会发生 q 次改变，具体来说， c 中的某两个元素会发生交换。（改变是永久的）

你需要对一开始的 c 和每次改变后的 c 分别求出它的价值对质数 P 取模的结果。

输入格式

第一行三个非负整数 n, q, P ，意义如题面所示。

第二行一个 32 位无符号整数 $seed$ ，表示随机数生成器的种子，用来生成括号序列 c ，你可以在选手文件夹下的 `bad/generator.cpp` 文件内了解如何生成 c 。

接下来 q 行，每行两个正整数 $x, y (1 \leq x, y \leq n)$ ，表示第 i 次改变将 c_x 和 c_y 交换。

输出格式

第一行一个整数，表示初始时 c 的价值对 P 取模的结果。

接下来 q 行每行一个整数，其中第 i 行表示第 i 次修改后的 c 的价值对 P 取模的结果。

样例 1 输入

```
6 3 998244853
2387462378
1 5
2 6
3 1
```

样例 1 输出

```
828
1292
```

1756

1524

样例 2 输入

详见选手文件夹下的 *bad/bad2.in* 文件。
这组样例满足测试点 2 的约束。

样例 2 输出

详见选手文件夹下的 *bad/bad2.ans* 文件。

样例 3 输入

详见选手文件夹下的 *bad/bad3.in* 文件。
这组样例满足测试点 4 的约束。

样例 3 输出

详见选手文件夹下的 *bad/bad3.ans* 文件。

数据范围与约束

本题采用捆绑测试，共 4 个测试点，你必须通过每个测试点中的所有数据才能获得该测试点的分数。

对于 100% 的数据，满足： $1 \leq n \leq 10^7$ 且 n 是偶数， $0 \leq q \leq 10^5$ ， $n < P \leq 10^9$ 且 P 是质数， $0 \leq seed < 2^{32}$ 。
具体的数据范围见下表：

测试点编号	n	特殊性质	分数
1	$\leq 10^3$	无	10
2	$\leq 10^5$	$P = 998244353$	35
3	$\leq 10^5$	无	20
4	$\leq 10^7$	无	35