

预览输出

试机

时间：1926 年 8 月 17 日 00:00 ~ 00:00:01

题目名称	数数	数树	鼠树
题目类型	传统型	传统型	传统型
可执行文件名	cuvelia.exe	voltississimo.exe	pastel.exe
输入文件名	cuvelia.in	voltississimo.in	pastel.in
输出文件名	cuvelia.out	voltississimo.out	pastel.out
每个测试点时限	1.0 秒	2.0 秒	3.0 秒
内存限制	512 MiB	1024 MiB	512 MiB
子任务数目	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	cuvelia.cpp	voltississimo.cpp	pastel.cpp
-----------	-------------	-------------------	------------

编译选项

对于 C++ 语言	-lm -Wl,--stack=536870912	-lm -Wl,--stack=1073741824	-lm -Wl,--stack=536870912
-----------	------------------------------	-------------------------------	------------------------------

数数 (cuvelia)

【题目描述】

Cuvelia 给了你一个长度为 n 的序列 $a_1 \cdots a_n$ 。对于所有 $k \in [1, n]$, Cuvelia 想让你选出序列中 k 个不同的下标 $i_1, i_2 \cdots i_k$ 使得 $\sum_{l=1}^k \sum_{r=l}^k |a_{i_l} - a_{i_r}|$ 最大, 并求出这个最大值。

【输入格式】

从文件 *cuvelia.in* 中读入数据。
第一行一个正整数 n 。
第二行 n 个整数表示 $a_1 \cdots a_n$ 。

【输出格式】

输出到文件 *cuvelia.out* 中。
输出 k 行, 每行一个非负整数, 第 i 行表示选择 i 个不同下标时的最大值。

【样例 1 输入】

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

【样例 1 输出】

```
1 0
2 17
3 34
4 53
```

【样例 2】

见选手目录下的 *cuvelia/cuvelia2.in* 与 *cuvelia/cuvelia2.ans*。

【子任务】

对于所有数据， $|a_i| \leq 10^6$ 。

测试点	n	特殊性质
1,2,3	$= 5$	
4,5	$= 10^2$	
6,7	$= 2,000$	
8,9	$= 299,999$	$a_i \in \{0,1\}$
10	$= 3 \times 10^5$	

数树 (voltississimo)

【题目描述】

Rasis 给了你一颗 n 个点的树，节点编号为 $1 - n$ ，Rasis 还给这棵树上的每条边 (u, v) 规定了一个方向。Rasis 想给这棵树上每个点分配一个权值 a_i ，满足 $a_1 \cdots a_n$ 组成一个 $1 - n$ 的排列且不存在一条从 u 指向 v 的边满足 $a_v = a_u + 1$ （注意从 u 指向 v 意味着这个约束是单向的）。Rasis 想知道这么分配权值的方案数对 998244353 取模的值。

两种分配方案不同，当且仅当存在一个节点的权值在两种方案中不相同。

【输入格式】

从文件 *voltississimo.in* 中读入数据。

第一行一个正整数 n 。

接下来 $n - 1$ 行每行两个整数 u, v ，表示树上一条从 u 指向 v 的边。

【输出格式】

输出到文件 *voltississimo.out* 中。

输出一行一个整数，表示不同的合法分配方案数对 998244353 取模的值。

【样例 1 输入】

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

【样例 1 输出】

```
1 10
```

【样例 2】

见选手目录下的 *voltississimo/voltississimo2.in* 与 *voltississimo/voltississimo2.ans*。

【子任务】

测试点	n	特殊性质
1,2	$= 5$	
3,4,5	$= 13$	
6,7	$= 18$	
8,9	$= 10^2$	
10,11,12	$= 300$	
13,14,15	$= 4,998$	树是一条链
16,17,18	$= 4,999$	树是外向树
19,20	$= 5,000$	

树是一条链的定义为：存在一个长度为 n 的排列 a ，使得对于所有整数 $i \in [1, n - 1]$ ，存在一条边由 a_i 指向 a_{i+1} 。

树是外向树的定义为：从点 1 出发沿有向边可以到达所有其他点。

鼠树 (pastel)

【题目描述】

Pastel 有一颗 n 个点的树，节点编号为 $1 - n$ ，树以编号为 1 的点为根。每个点都有一个颜色，为黑色或白色。初始时，只有根节点的颜色为黑色，其他的点颜色均为白色。每个点 i 还有一个权值 w_i ，初始时均为 0。

Pastel 定义一个点的深度为这个点与根节点之间的路径经过的边数。对于一个点 i ，Pastel 称点 i 到根的路径上深度最大的黑点为点 i 的**归属点**（可以是这个点本身）。对于一个黑色点 j ，Pastel 称所有**归属点**为点 j 的点 i 为点 j 的**管辖点**。

现在 Pastel 要对这棵树进行 m 次操作，有以下几种：

- 1 k 表示询问节点 k 的权值。
- 2 $k\ w$ 表示将黑色点 k 的所有**管辖点**的权值加上 w 。
- 3 k 表示询问节点 k 子树中所有点的权值和。
- 4 $k\ w$ 表示对节点 k 子树内的所有黑色点 l ，把 l 的所有**管辖点**的权值加上 w （既对子树 k 内所有黑点 l 进行操作 2 1 w ）。
- 5 k 表示把节点 k 的颜色变为黑色。
- 6 k 表示把节点 k 的颜色变为白色。

每个点的管辖点集合或归属点都会随着操作实时变化。Pastel 想要知道每次操作 1 与 3 的答案。由于答案可能很大，你只需要回答答案对 2^{32} 取模（既 unsigned int 数据类型的自然溢出）的结果。

【输入格式】

从文件 `pastel.in` 中读入数据。

第一行两个正整数 $n\ m$ 。

接下来 $n - 1$ 行每行一个正整数 v_i ，第 i 行表示节点 i 的父节点为 v_i 。

接下来 m 行每行描述一次操作。

【输出格式】

输出到文件 `pastel.out` 中。

对每次类型 1 或 3 的操作，输出一行一个整数表示取模后的答案。

【样例 1 输入】

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

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

【样例 1 输出】

```
1 0
2 2
3 4
4 16
5 10
```

【样例 2】

见选手目录下的 *pastel/pastel2.in* 与 *pastel/pastel2.ans*。

【子任务】

对于所有数据，点 1 的颜色在操作过程中永远为黑色，操作5与6的进行一定会导致某个点的颜色改变。

操作中，有 $k \in [1, n]$ 且合法， $w \in [0, 10^9]$ 。

测试点	n	m	操作类型	树是一条链	颜色固定
1,2	$= 10^3$	$= 3,000$	1,2,3,4,5,6	否	否
3,4,5	$= 299,996$	$= 5 \times 10^5$	1,2,3,4,5		是
6,7,8,9	$= 299,997$		1,2,4,5		否
10,11,12	$= 299,998$		1,2,3,4,5,6	是	
13,14,15	$= 299,999$		1,2,3,4,5	否	
16,17,18,19,20	$= 3 \times 10^5$		1,2,3,4,5,6		

树是一条链的定义为：对于所有整数 $i \in [1, n - 1]$ ，点 i 与点 $i + 1$ 之间有边相连。
颜色固定的定义为：进行一次操作1或2或3或4后，不会再出现操作5。