

2026 全国青少年信息学奥林匹克竞赛

CCF NOI 2026

第二试

时间：2026 年 7 月 22 日 08:00 ~ 13:00

题目名称	中位数	木棉	彩虹树
题目类型	传统型	传统型	传统型
目录	median	kapok	rainbow
每个测试点时限	1.0 秒	2.0 秒	1.0 秒
内存限制	1024 MiB	1024 MiB	1024 MiB
测试点数目	20	25	25
测试点是否等分	是	是	是
预测试点数目	20	25	25

提交程序源文件名

对于 C++ 语言	median.cpp	kapok.cpp	rainbow.cpp
-----------	------------	-----------	-------------

编译选项

对于 C++ 语言	-O2 -std=c++14 -static
-----------	------------------------

注意事项（请仔细阅读）

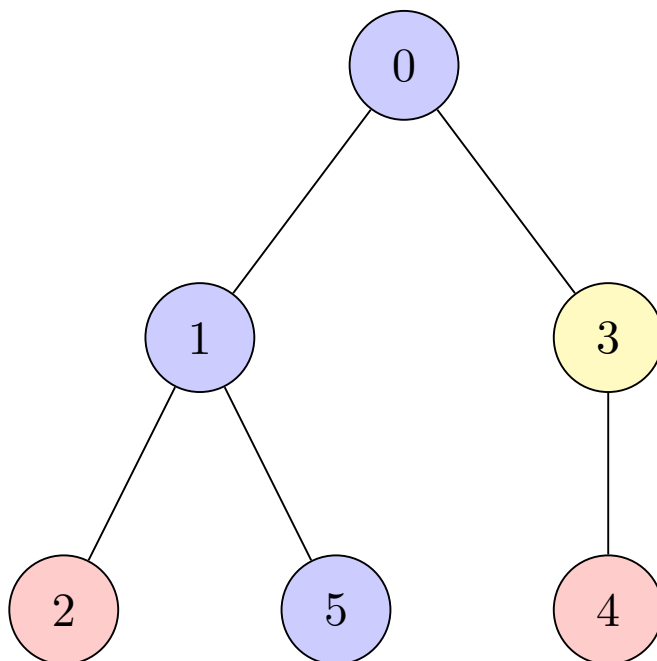
- 赛后正式测试时将以选手留在题目目录下的程序源文件为准。对程序源文件以外文件的修改不会影响评测结果。
- 选手提交的程序源文件大小不得超过 100 KiB。
- 程序可使用的栈空间内存限制与题目的内存限制一致。
- 禁止在提交的程序源文件中改变编译器参数（如使用 `#pragma` 命令），禁止使用系统结构相关指令（如内联汇编）或其他可能造成不公平的方法。
- 因违反上述规定而出现的问题，申诉时一律不予受理。
- 选手可使用快捷启动页面中的工具 selfEval 进行自测。选手需将待测程序的源文件置于相应题目目录下。每次自测时可选择全部或部分题目进行自测。注意：自测有次数限制，且自测结果仅供选手测试参考，不作为最终正式成绩。

彩虹树 (rainbow)

【题目描述】

传说在夜之国中，有一棵包含 n 个结点的彩虹树。彩虹树是一棵有根树，结点编号为 $0 \sim n-1$ ，其中结点 0 是彩虹树的树根，结点 i ($1 \leq i < n$) 的父亲结点为 f_i 。

彩虹树上的每个结点都能显现出任意颜色。颜色共有无穷多种，但彩虹树的**缤纷度**只由每棵子树中出现过的**不同颜色的数量**决定，而与具体颜色无关。具体地，设以结点 i ($0 \leq i < n$) 为根的子树中共有 c_i 种不同的颜色的结点，则彩虹树的缤纷度可以用序列 $[c_0, c_1, \dots, c_{n-1}]$ 表示。例如，在下图中，结点 0, 1, 5 为蓝色，结点 2, 4 为红色，结点 3 为黄色，因此彩虹树的缤纷度为 $[3, 2, 1, 2, 1, 1]$ 。



云层的变化会导致彩虹树的色彩显现出某种特定的规律。具体地，每种规律可以由一个对应的结点子集 $S \subseteq \{1, 2, \dots, n-1\}$ 来描述：集合 S 中的每个结点 u 的颜色都与某个祖先颜色相同。形式化地，对于所有 $u \in S$ ，均存在 u 的祖先 p ($p \neq u$)，满足 u 与 p 颜色相同。尽管受到上述规律的制约，彩虹树仍能呈现出不同的色彩，形成不同的缤纷度。记满足规律 S 的彩虹树的缤纷度的**种类数**为 w_S ，其中两种缤纷度不同当且仅当两个序列中至少有一个元素不同。

请计算，对于所有可能的 2^{n-1} 种规律，满足每种规律的彩虹树的缤纷度的种类数之和，即 $\sum_{S \subseteq \{1, 2, \dots, n-1\}} w_S$ 。由于答案可能较大，只需求出答案对 998,244,353 取模后的结果。

【实现细节】

选手不需要，也不应该实现 `main` 函数。

选手需要确保提交的程序源文件包含头文件 `rainbow.h`，即在程序开头加入以下代码：

```
1 #include "rainbow.h"
```

选手需要在提交的程序源文件 `rainbow.cpp` 中实现以下函数：

```
1 int rainbow(int c, int n, std::vector<int> f);
```

- c, n 分别表示测试点编号与彩虹树的结点数。 $c = 0$ 表示该测试点为样例。
- f 是一个长度为 n 的序列，其中 $f_0 = 0$ ， f_i ($1 \leq i < n$) 表示结点 i 的父亲结点。
- 该函数需要返回满足每种规律的彩虹树的缤纷度的种类数之和对 998,244,353 取模后的结果。
- 对于每个测试点，该函数会被评测程序调用恰好一次。

本试题目录下的 `template_rainbow.cpp` 是提供的示例代码，选手可参考并实现自己的代码。

【测试程序方式】

选手可以在本题目录下使用如下命令编译得到可执行文件：

```
1 g++ grader.cpp rainbow.cpp -o rainbow -O2 -std=c++14 -static
```

对于编译得到的可执行文件 `rainbow`：

- 可执行文件将从标准输入读入以下格式的数据：
 - 第一行包含两个非负整数 c, n 。
 - 第二行包含 $n - 1$ 个非负整数 $f_1, f_2, \dots, f_{n-1}$ 。
- 可执行文件将输出以下格式的数据至标准输出：
 - 输出一行一个非负整数表示 `rainbow` 函数的返回值。

【样例 1 输入】

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

【样例 1 输出】

```
1 8
```

【样例 1 解释】

- 满足规律 $S = \emptyset$ 的彩虹树的缤纷度共有 $[1, 1, 1], [2, 1, 1], [3, 1, 1]$ 三种;
- 满足规律 $S = \{1\}$ 的彩虹树的缤纷度共有 $[1, 1, 1], [2, 1, 1]$ 两种;
- 满足规律 $S = \{2\}$ 的彩虹树的缤纷度共有 $[1, 1, 1], [2, 1, 1]$ 两种;
- 满足规律 $S = \{1, 2\}$ 的彩虹树的缤纷度有 $[1, 1, 1]$ 一种。

因此答案为 $3 + 2 + 2 + 1 = 8$ 。

【样例 2 输入】

```
1 0 6
2 0 1 0 3 1
```

【样例 2 输出】

```
1 279
```

【样例 3】

见选手目录下的 *rainbow/rainbow3.in* 与 *rainbow/rainbow3.ans*。
该样例满足测试点 3, 4 的约束条件。

【样例 4】

见选手目录下的 *rainbow/rainbow4.in* 与 *rainbow/rainbow4.ans*。
该样例满足测试点 5 ~ 7 的约束条件。

【样例 5】

见选手目录下的 *rainbow/rainbow5.in* 与 *rainbow/rainbow5.ans*。
该样例满足测试点 8, 9 的约束条件。

【样例 6】

见选手目录下的 *rainbow/rainbow6.in* 与 *rainbow/rainbow6.ans*。
该样例满足测试点 10 ~ 12 的约束条件。

【数据范围】

对于所有测试数据，均有：

- $1 \leq n \leq 200$;
- 对于所有 $1 \leq i < n$ ，均有 $0 \leq f_i < i$ 。

测试点编号	$n \leq$	特殊性质
1	4	无
2	8	
3,4	16	
5 ~ 7	50	
8,9	10^2	B
10 ~ 12		无
13 ~ 15	150	
16	200	A
17 ~ 19		B
20 ~ 25		无

特殊性质 A：对于所有 $1 \leq i < n$ ，均有 $f_i = i - 1$ 。

特殊性质 B：对于所有 $0 \leq i < n$ ，至多存在两个 j 满足 $f_j = i$ 。