

NOI 2025 模拟赛

题目名称	回文	保险丝	冷门肘法信仰玩家
题目类型	传统型	传统型	传统型
目录	palindrome	fuse	player
可执行文件名	palindrome	fuse	player
输入文件名	palindrome.in	fuse.in	player.in
输出文件名	palindrome.out	fuse.out	player.out
提交文件名	palindrome.cpp	fuse.cpp	player.cpp
时间限制	1.5 秒	1.0 秒	1.0 ~ 4.5 秒
内存限制	1024 MiB	512 MiB	128 ~ 512 MiB
子任务数目	5	20	10
测试点是否等分	否	是	否
编译选项	-O2 -std=c++14		

【注意事项（请仔细阅读）】

1. 选手提交的源程序请直接放在个人目录下，无需建立子文件夹；
2. 文件名（包括程序名和输入输出文件名）必须使用英文小写。
3. C++ 中函数 `main()` 的返回值类型必须是 `int`，值必须为 0。
4. 对于因未遵守以上规则对成绩造成的影响，相关申诉不予受理。
5. 若无特殊说明，结果比较方式为忽略行末空格、文末回车后的全文比较。
6. 程序可使用的栈空间大小与该题内存空间限制一致。
7. 在终端中执行命令 `ulimit -s unlimited` 可将当前终端下的栈空间限制放大，但你使用的栈空间大小不应超过题目限制。
8. 若无特殊说明，每道题的代码大小限制为 100KB。
9. 若无特殊说明，输入与输出中同一行的相邻整数、字符串等均使用一个空格分隔。
10. 输入文件中可能存在行末空格，请选手使用更完善的读入方式（例如 `scanf` 函数）避免出错。
11. 直接复制 PDF 题面中的多行样例，数据将带有行号，建议选手直接使用对应目录下的样例文件进行测试。
12. 使用 `std::deque` 等 STL 容器时，请注意其内存空间消耗。

13. 请务必使用题面中规定的的编译参数，保证你的程序在本机能够通过编译。此外不允许在程序中手动开启其他编译选项，一经发现，本题成绩以 0 分处理。

14. 统一评测时采用的机器配置为：12th Gen Intel(R) Core(TM) i7-12700 2.10 GHz，内存 16GB。上述时限以此配置为准。

15. 评测在 Windows 11 下进行，使用 LemonLime 进行评测。如果对此条以及 14 条中的机器配置有疑问，请及时询问。

16. 不保证题目严格按照难度顺序排序。

回文 (palindrome)

【题目描述】

对于一个长为 len 的, 取值范围为 $[1, m]$ 的序列 S , 称一个长度为 m 的, 取值范围为 $[1, m]$ 的序列 T 是好的, 当且仅当对于所有 $i \in [1, len]$, 都满足 $T_{S_i} = S_{len+1-i}$ 。 $f(S)$ 表示有多少个好的序列。

给出一个长为 n 的串 A , 再给出值域 m , 要你求出 A 的所有非空子串的 f 之和, 即 $\sum_{i=1}^n \sum_{j=i}^n f(A[i, j])$ 。对 998244353 取模。

【输入格式】

从文件 *palindrome.in* 中读入数据。

第一行输入两个正整数 n 与 m , 即串长与值域大小。

第二行输入 n 个正整数, 表示 n 个元素的值。

【输出格式】

输出到文件 *palindrome.out* 中。

一个数, 表示答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 272
```

【样例 2 输入】

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

【样例 2 输出】

```
1 67
```

【样例 3 输入】

```
1 10 4
2 4 2 3 4 4 1 2 3 1 1
```

【样例 3 输出】

```
1 914
```

【样例 4】

见附件下的 *palindrome/palindrome4.in* 与 *palindrome/palindrome4.ans*。
该样例满足子任务 1 的限制条件。

【样例 5】

见附件下的 *palindrome/palindrome5.in* 与 *palindrome/palindrome5.ans*。
该样例满足子任务 2 的限制条件。

【样例 6】

见附件下的 *palindrome/palindrome6.in* 与 *palindrome/palindrome6.ans*。
该样例满足子任务 3 的限制条件。

【样例 7】

见附件下的 *palindrome/palindrome7.in* 与 *palindrome/palindrome7.ans*。
该样例满足子任务 4 的限制条件。

【样例 8】

见附件下的 *palindrome/palindrome8.in* 与 *palindrome/palindrome8.ans*。
该样例满足子任务 5 的限制条件。

【测试点约束】

本题使用捆绑测试，对于所有数据，满足 $1 \leq n, m \leq 5 \times 10^6, A_i \in [1, m]$ 。

子任务编号	特殊限制	分值
1	$n, m \leq 6$	10
2	$n, m \leq 300$	20
3	$n, m \leq 6000$	20
4	$n, m \leq 4 \times 10^5$	30
5	无特殊限制	20

保险丝 (fuse)

【题目描述】

给定一棵 n 个点的有根树，根是 1 号结点。

定义两个点集 S_1, S_2 的距离为从两个集合分别选出一个点，能得到两点间距离的最小值，即 $\text{dist}(S_1, S_2) = \min_{u \in S_1 \wedge v \in S_2} \text{dist}(u, v)$ ，其中 $\text{dist}(u, v)$ 是点 u, v 间的距离。

定义 $\text{path}(u, v)$ 是 u 到 v 的简单路径上的所有点组成的集合， \mathcal{L} 是所有叶子组成的集合。

对于固定正整数 u ，定义满足如下条件的结点 v 构成 u 的半邻域 $\mathring{U}(u)$ ：

- v 在 u 子树内；
- $\text{dist}(u, v) \leq \text{dist}(\text{path}(1, v), \mathcal{L})$ 。

即 u 的半邻域 $\mathring{U}(u)$ 包含 u 的子树内所有满足到 u 的距离不大于它到根的路径上任意一点离最近叶子节点的距离的点。

进而定义：

$$f(x) = \sum_{u \in \mathring{U}(x)} \prod_{v \in \text{subtree}(u) \cap \mathring{U}(x)} F_{\deg v}$$

其中 $\text{subtree}(u)$ 是 u 子树中所有点组成的集合， $\deg u$ 是 u 的度数（与 u 有连边的点的数量）， F 是 Fibonacci 数列：

$$F_n = \begin{cases} 1 & n \leq 2 \\ F_{n-1} + F_{n-2} & n \geq 3 \end{cases}$$

即 $f(x)$ 对应 x 的半邻域中点对 x 的贡献之和。而一个点 u 对 x 的贡献的计算方式为：取出每个 u 子树内处在 x 半邻域中的点 v ，若 v 的度数为 d ，则将 u 的贡献乘上 F_d ，所有 u 的贡献之和为结果。

你需要求出 $f(1), f(2), \dots, f(n)$ 的值。为减少输出量，你只需要输出它们模 994007158 后的异或和，即 $\bigoplus_{x=1}^n (f(x) \bmod 994007158)$ 即可。

【输入格式】

从 *fuse.in* 中读入数据。

第一行一个正整数 n 表示点数。

第二行 $n-1$ 个正整数，第 i 个正整数代表了结点 $i+1$ 的父亲结点。

【输出格式】

输出到文件 ***fuse.out*** 中。

一行，表示 $\bigoplus_{x=1}^n (f(x) \bmod 994007158)$ 的值。

【样例 1 输入】

```
1 7
2 1 1 2 2 3 3
```

【样例 1 输出】

```
1 8
```

【样例 1 解释】

第一组数据中 f 在 $1 \dots 7$ 处的取值：8, 2, 2, 1, 1, 1, 1。

【样例 2 输入】

```
1 14
2 1 2 3 3 2 6 6 6 9 9 10 11 12
```

【样例 2 输出】

```
1 25
```

【样例 2 解释】

第二组数据中 f 在 $1 \dots 14$ 处的取值：4, 17, 2, 1, 1, 8, 1, 1, 4, 2, 1, 1, 1, 1。

【样例 3】

见附件下的 ***fuse/fuse3.in*** 与 ***fuse/fuse3.ans***。

该样例满足测试点 1 ~ 2 的限制条件。

【样例 4】

见附件下的 *fuse/fuse4.in* 与 *fuse/fuse4.ans*。
该样例满足测试点 3 ~ 4 的限制条件。

【样例 5】

见附件下的 *fuse/fuse5.in* 与 *fuse/fuse5.ans*。
该样例满足测试点 5 的限制条件。

【样例 6】

见附件下的 *fuse/fuse6.in* 与 *fuse/fuse6.ans*。
该样例满足测试点 6 ~ 8 的限制条件。

【样例 7】

见附件下的 *fuse/fuse7.in* 与 *fuse/fuse7.ans*。
该样例满足测试点 9 的限制条件。

【样例 8】

见附件下的 *fuse/fuse8.in* 与 *fuse/fuse8.ans*。
该样例满足测试点 10 ~ 11 的限制条件。

【样例 9】

见附件下的 *fuse/fuse9.in* 与 *fuse/fuse9.ans*。
该样例满足测试点 12 ~ 13 的限制条件。

【样例 10】

见附件下的 *fuse/fuse10.in* 与 *fuse/fuse10.ans*。
该样例满足测试点 14 ~ 15 的限制条件。

【样例 11】

见附件下的 *fuse/fuse11.in* 与 *fuse/fuse11.ans*。
该样例满足测试点 16 ~ 18 的限制条件。

【样例 12】

见附件下的 *fuse/fuse12.in* 与 *fuse/fuse12.ans*。
该样例满足测试点 19 ~ 20 的限制条件。

【测试点约束】

对于全部数据， $2 \leq n \leq 10^6$ ，每个非根结点父亲的编号小于它的编号。

测试点编号	$n \leq$	特殊限制
1 ~ 2	500	/
3 ~ 4	3000	/
5	5000	A
6 ~ 8	1000000	A
9	500000	B
10 ~ 11	500000	C
12 ~ 13	1000000	C
14 ~ 15	100000	/
16 ~ 18	500000	/
19 ~ 20	1000000	/

特殊限制	具体内容
A	保证树是一条链
B	保证树是一张菊花图
C	保证树中除根以外每个节点的度数均 ≥ 3

冷门肘法信仰玩家 (player)

【题目背景】

孩子们，我被削废了。what can I say?



【题目描述】

小 F 是一位坚持练习冷门超模角色牢眼肘法的食力冷门肘法信仰玩家，他将他的一局游戏分成了若干个循环的阶段，控场，抓人，控场，抓人……

在不断练习肘法后，已经从普通的冷门信仰玩家荣升为冷门肘法信仰玩家的小 F 将他需要控场的目标抽象成了一颗树上的 n 个节点，因为他将游戏的打法神奇地找出了规律，所以他的打法是有周期性的，比如他控场的顺序是一个长度为 n 的排列 a 顺时针组成的环，你可以认为在环上 a_n 顺时针方向下一个数就是 a_1 。

他的计划是将这一系列控场目标组成的环分成 k 段，每一段的间隙去抓人。

对于每一次控场，他可以消耗 1 点能量修建机关墙封住边 (u, v) 使其不能通行。而小 F 认为一次控场为**完美的控场**当且仅当你划分的段中的点两两之间的路径上不存在能通行的边并且**消耗的能量最小**，并且他认为一次控场的**超模度**为此时被封住的边的总数。

作为冷门信仰玩家的小 F 自然想让 k 次控场都是**完美的控场**的前提下各段的**超模度**之和最大，不过他现在因为土豆服务器太卡了红如温导致无法仔细思考，所以请求你写一个程序帮他算一算。

形式化题意：

给你一个 n 个点，边权为 1 的树 T 和一个长度为 n 的环 a ，保证环上没有两个数相等。

对于 T 中一个非空的顶点的集合 S ，令 $f(S)$ 为包含 S 中每个节点的最小连通子树的最小边数，即虚树的大小。

定义环上一个子段 (l, r) 对应的点集 $S(l, r)$ 为：

$$S(l, r) = \begin{cases} [l, r] \cap \mathbb{Z} & l \leq r \\ ([l, n] \cup [1, r]) \cap \mathbb{Z} & l > r \end{cases}$$

你要把环分割为 k 个子段，定义一个在环上按顺时针方向端点依次为 l, r 的子段的权值为 $f(S(l, r))$ ，求所有划分中 k 个子段权值之和的最大值。

【输入格式】

从文件 **player.in** 中读取数据。

一共 $n + 1$ 行。

第 1 行 3 个整数 id, n, k ，表示测试点编号（样例编号为 0），排列及树的大小和小 F 想要分的段数。

第 2 行一共 n 个数 $a_{1 \dots n}$ ，表示从 a_1 所在位置开始顺时针依次为哪个数，保证 a 是一个 $1 \sim n$ 的排列。

第 3 $\sim n + 1$ 行每行 2 个整数 u, v ，表示树的一条边 (u, v) 。

【输出格式】

输出到文件 **player.out** 中。

一行一个整数表示所求答案。

【样例 1 输入】

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

【样例 1 输出】

```
1 5
```

【样例 1 解释】

一种可行的方案是划分为 $\{5, 2\}$ 和 $\{4, 3, 1\}$ 两段，超模度分别为 3 和 2，此时超模度之和有最大值 5。

【样例 2 输入】

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

【样例 2 输出】

```
1 19
```

【样例 2 解释】

一种可行的方案是划分为 $\{7, 9, 6, 11\}, \{2, 10, 3, 1\}, \{12, 8\}, \{4, 5\}$ 四段，超模度分别为 $8, 3, 5, 3$ ，此时超模度之和有最大值 19。

【样例 3 输入】

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

```
13 15 12
14 16 3
15 17 8
16 18 10
17 19 1
18 20 13
19 12 5
20 2 13
21 2 1
```

【样例 3 输出】

```
1 34
```

【样例 4】

见附件下的 *player/player4.in* 与 *player/player4.ans*。
该组样例满足子任务 1 的约束条件。

【样例 5】

见附件下的 *player/player5.in* 与 *player/player5.ans*。
该组样例满足子任务 2 的约束条件。

【样例 6】

见附件下的 *player/player6.in* 与 *player/player6.ans*。
该组样例满足子任务 3 的约束条件。

【样例 7】

见附件下的 *player/player7.in* 与 *player/player7.ans*。
该组样例满足子任务 4 的约束条件。

【样例 8】

见附件下的 *player/player8.in* 与 *player/player8.ans*。
该组样例满足子任务 5 的约束条件。

【样例 9】

见附件下的 *player/player9.in* 与 *player/player9.ans*。
该组样例满足子任务 6 的约束条件。

【样例 10】

见附件下的 *player/player10.in* 与 *player/player10.ans*。
该组样例满足子任务 7 的约束条件。

【样例 11】

见附件下的 *player/player11.in* 与 *player/player11.ans*。
该组样例满足子任务 8 的约束条件。

【样例 12】

见附件下的 *player/player12.in* 与 *player/player12.ans*。
该组样例满足子任务 9 的约束条件。

【样例 13】

见附件下的 *player/player13.in* 与 *player/player13.ans*。
该组样例满足子任务 10 的约束条件。

【测试点约束】

温馨提示：请注意常数因子对程序效率的影响，保证时空限制至少是 std 的 1.5 倍。
对于 100% 的数据， $1 \leq n \leq 10^5, 2 \leq k \leq n, 1 \leq u, v, a_i \leq n$ 。
本题开启捆绑测试和合理的子任务依赖，**请注意不同测试点的不同时空限制。**

子任务编号	分值	$n \leq$	$k \leq$	特殊性质	子任务依赖	时间限制	空间限制
1	3	2000	2	无	无	2s	128MB
2	3	100	10	无	无	2s	128MB
3	9	700	25	无	2	2s	128MB
4	15	1500	n	无	3	2s	256MB
5	5	5000	n	无	4	3s	256MB
6	13	50000	n	$nk \leq 250000$	4	3s	128MB
7	17	100000	n	$nk \leq 2000000$	6	3s	128MB
8	7	25000	n	无	5	4.5s	512MB
9	10	50000	n	无	8	4.5s	512MB
10	18	100000	n	无	9	4.5s	128MB