

# 树上连通块

[树上连通块 - 题目 - Daimayuan Online Judge](#)

## 题目简介

给定一个点，求出包含该点的连通块的个数，答案对 $m(2 \leq m \leq 10^9)$ 取模。 $1 \leq n \leq 10^5$

## solve

1. 先考虑一个最简单的dp问题。只是求一个点。

1. 定义状态 $f_i$ 表示的是，以 $i$ 为根的子树中连通块的个数（不包含节点数为0的情况）。

2. 状态转移方程：

$$f_i = (f_{son_1} + 1) \times (f_{son_2} + 1) \times \dots \times (f_{son_i} + 1) \quad (1)$$

2.

3. 考虑要求所有点：

1. 考虑简单的换根dp.，每次换根计算出更新的dp值

1. 暴力。 $O(deg * deg) \rightarrow O(n^2)$

2. 对父亲的dp值直接 除法。（ $m$ 不一定是质数，无法计算出逆元）

换根dp优化实现方法：

1. 维护儿子的前缀积，后缀积。局部修改问题。直接考虑前后缀的情况。

下面是一些困惑：

1. 换根dp的顺序是什么？

随意找一个点作为根，发现不能处理父亲的情况。作为子树的贡献情况。换根的前提是，待选根的父亲作为子树的方案贡献已经求 出来。

2. 换根的过程中，节点的dp要出现什么变化？

节点的dp不需要变化。再开一个dp2用来表示，该节点的父亲作为其余节点构成的子树的父亲时，其子树的连通块种类（不包括0的情况）。

关于dp2的转移方程：

1. 对于深度为1的节点，就是兄弟节点的 $(dp + 1)$ 相乘。为什么要加一（因为可以考虑不选）。

2. 对于深度不为1的节点，就是兄弟节点乘上父亲的 $(dp2 + 1)$ 。

## code

```
1  #include<bits/stdc++.h>
2  using namespace std;
3  using ll = long long;
4
5  const int N = 1E5 + 10;
6  int n , mod;
7  vector<int> son[N];
8  ll dp[N] , dp2[N] , ans[N];
9
10 void dfs(int u) {
```

```

11     dp[u] = 1;
12     for (auto v : son[u]) {
13         dfs(v);
14         dp[u] = dp[u] * (dp[v] + 1) % mod;
15     }
16 }
17
18 void dfs2(int u) {
19     int m = son[u].size(); if (m == 0) return ;
20     static ll pre[N] , suf[N];
21
22     pre[0] = 1;
23     for (int i = 1; i <= m; i++) {
24         int v = son[u][i - 1];
25         pre[i] = pre[i - 1] * (dp[v] + 1) % mod;
26     }
27     suf[m + 1] = 1;
28     for (int i = m; i >= 1; i--) {
29         int v = son[u][i - 1];
30         suf[i] = suf[i + 1] * (dp[v] + 1) % mod;
31     }
32     for (int i = 0; i < m; i++) {
33         int v = son[u][i];
34         dp2[v] = pre[i] * suf[i + 2] % mod;
35         if (u != 1) dp2[v] = dp2[v] * (dp2[u] + 1) % mod;
36     }
37     for (auto v : son[u]) {
38         ans[v] = (dp2[v] + 1) * dp[v] % mod;
39         dfs2(v);
40     }
41 }
42
43 int main()
44 {
45     ios::sync_with_stdio(false);
46     cin.tie(0);
47
48     cin >> n >> mod;
49     for (int i = 2; i <= n ; i++) {
50         int x; cin >> x;
51         son[x].push_back(i);
52     }
53     dfs(1);
54     dfs2(1);
55     ans[1] = dp[1];
56     for (int i = 1; i <= n; i++) {
57         cout << ans[i] << '\n';
58     }
59 }
60
61 /* stuff you should look for
62 * int overflow, array bounds
63 * special cases (n=1?)
64 * do smth instead of nothing and stay organized
65 * WRITE STUFF DOWN

```

```
66 * DON'T GET STUCK ON ONE APPROACH
67 */
```

## 生长思考

1. 换根dp也有一定的模式：

1. 比方说，专门开一个dp2表示节点的父亲作为其余节点构成的根时，所定义的指标函数的解。