Problem - K - Codeforces

简介

给定一棵树:

根节点处有无数支军队。不断地,可以使用一个单位地时间,将一支军队转移到相邻地节点。问至少要 花费多少地时间,使得每一个节点都被军队遍历过。

solve

1. 考察最终的答案。最终每一支军队都将会分布在叶子节点处。

问几个问题:

探究解空间:

1. 一颗树的节点会被怎样遍历?

考虑每一个儿子子树的遍历情况。

- 1. 没有军队停留在树的叶子下:
- 2. 至少一个军队进入:军队停留在叶子上。
- 3. 如果至少一支军队停留,那么每一支军队都将停留在子树的叶子节点下。

状态定义:

 $f_{u,0/1}$ 分别表示,有没有下派军队情况下,处理以u为根的子树的最小代价。

状态转移方程为:

$$f_{v,0} = \sum (f_{v,0} + 2)$$

 $f_{u,1}$ 若干 $f_{v,1},f_{v,0}$ 的组合。同时注意至少有一个 $f_{v,1}$

code (dfs版本)

```
1 #include<bits/stdc++.h>
   using namespace std;
 3
    typedef long long 11;
 5 const int oo = 0x0fffffff;
   const int N = 1E6 + 10;
 7
 8
    vector<int> e[N];
9
    int dp[N][2];
10
    int dep[N];
11
12
13 void dfs(int u) {
14
        dp[u][0] = 0;
        dp[u][1] = dep[u];
15
16
        for (auto v : e[u]) {
```

```
17
            dep[v] = dep[u] + 1;
18
            dfs(v);
            dp[u][1] = min({dp[u][1] + dp[v][0] + 2, dp[u][0] + dp[v][1], dp[u]}
19
    [1] + dp[v][1];
20
            dp[u][0] += dp[v][0] + 2;
21
        }
22
    }
23
24
25
    void work(int testNo)
26
    {
27
28
        int n; cin >> n;
        for (int i = 1; i \le n; i++) e[i].clear();
29
30
        for (int i = 2; i \le n; i++) {
31
            int x;
            cin >> x;
32
33
            e[x].push_back(i);
34
        }
35
        dfs(1);
        cout << "Case #" << testNo << ": " << min(dp[1][1] , dp[1][0]) <<</pre>
36
    '\n';
37
    }
38
39
40
    int main()
41
    {
42
        ios::sync_with_stdio(false);
43
        cin.tie(0);
44
45
        int t; cin >> t;
46
        for (int i = 1; i \le t; i++)work(i);
47
    }
```

code2(常数优化版本)

```
1 #include<bits/stdc++.h>
 2
    using namespace std;
    typedef long long 11;
 4
    const int oo = 0x0fffffff;
 5
    const int N = 1E6 + 10;
 6
 7
    int dp[N][2];
8
    int dep[N];
9
    int fa[N];
    void work(int testNo)
10
11
12
        int n; cin >> n;
13
        for (int i = 2; i <= n; i++) {
14
15
            cin >> fa[i];
            dep[i] = dep[fa[i]] + 1;
16
17
18
        for (int i = 1; i \le n; i++) {
19
            dp[i][0] = 0;
```

```
dp[i][1] = dep[i];
21
      }
22
      for (int v = n; v > 1; v--) {
           int u = fa[v];
23
           dp[u][1] = min({dp[u][1] + dp[v][1] , dp[u][1] + dp[v][0] + 2, dp[u]}
24
    [0] + dp[v][1]});
25
          dp[u][0] += dp[v][0] + 2;
26
27
       cout << "Case #" << testNo << ": " << dp[1][1] << '\n';</pre>
28 }
29
30 int main()
31 {
32
       ios::sync_with_stdio(false);
33
       cin.tie(0);
34
35
       int t; cin >> t;
36
       for (int i = 1; i \le t; i++)work(i);
37 }
```