

# 换根dp

---

## 人员

---

王梓同、蔡云翔、石宇赫、李佳声、王崇宇、窦浩轩 到课

于潇涵 请假

## 作业检查

---

王梓同 完成

蔡云翔 完成

石宇赫 完成

李佳声 完成

窦浩轩 完成

王崇宇 上周请假

## 作业

---

<https://www.luogu.com.cn/contest/164912>

A、B、C 题目必做

<https://www.luogu.com.cn/contest/162467>

刷题题单要求同学们课下进行练习

## 课堂表现

---

同学们做题时不画图考虑，细节考虑不全面。同学们以后做题一定要养成画图的习惯

## 课堂内容

---

P3478 [POI2008] STA-Station

考虑以每个点为根，做一遍dfs，求一下 这个点为根时到所有点的距离之和  
一共  $n$  个点可以为根，以每个点为根时dfs的过程是 $O(n)$ 的  
所以这种做法的时间复杂度是  $n * O(n)$ ，即  $O(n^2)$   
时间复杂度不符合题目要求，考虑优化

考虑以1这个点为根，先做一遍dfs，维护一个s数组和一个f数组  
s数组维护的是以这个点为根时子树中点的数量  
f数组维护的是以这个点为根时子树中所有点的总深度之和

这样，可以用 $O(n)$ 的时间做一遍dfs，求得f[1]代表以1为根时子树中所有点的总深度之和  
但是，此时的f[2], f[3]...代表的分别是以2, 3为根时子树中的总深度，所以要做第二遍dfs来更新f[2], f[3]...

核心转移： $f[i] += (f[u] - f[i] - s[i]) + (n - s[i])$   
其中  $(f[u] - f[i] - s[i])$  代表的是右边部分的点到u的总深度之和  
 $(n - s[i])$  代表的是右边部分所有的点从u到i的距离

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e6 + 5;
vector<int> vec[maxn];
LL f[maxn];
int s[maxn];

void dfs1(int u, int fa) {
    s[u] = 1;
    for (int i : vec[u]) {
        if (i == fa) continue;
        dfs1(i, u);
        s[u] += s[i];
        f[u] += f[i] + s[i];
    }
}

void dfs2(int u, int fa, int n) {
    for (int i : vec[u]) {
        if (i == fa) continue;
        f[i] += (f[u] - f[i] - s[i]) + (n - s[i]);
        dfs2(i, u, n);
    }
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n-1; ++i) {
        int u, v; cin >> u >> v;
        vec[u].push_back(v), vec[v].push_back(u);
    }

    dfs1(1, -1);
    dfs2(1, -1, n);

    LL res = f[1];
    int id = 1;
    for (int i = 2; i <= n; ++i) {
        if (f[i] > res) {
            res = f[i];
            id = i;
        }
    }
    cout << id << endl;
    return 0;
}

```

## CF1882D Tree XOR

易知：以某个点为根时，根节点的值一定不需要改动，一定是把根节点以下所有点的权值变的跟根节点相同

因为如果修改了根节点(比如异或5)，同时又会把根节点以下所有点全部修改(同时都异或5)  
那么所有点都异或5就等同于都不异或5

核心转移： $f[i] += f[u] - f[i] - s[i] * (a[i] ^ a[u]) + (n - s[i]) * (a[i] ^ a[u])$

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int w[maxn], s[maxn];
LL f[maxn];

void dfs1(int u, int fa) {
    s[u] = 1;
    for (int i : vec[u]) {
        if (i == fa) continue;
        dfs1(i, u);
        s[u] += s[i];
        f[u] += f[i] + (LL)s[i]*(w[u]^w[i]);
    }
}

void dfs2(int u, int fa, int n) {
    for (int i : vec[u]) {
        if (i == fa) continue;
        f[i] += (f[u]-f[i]-(LL)s[i]*(w[u]^w[i])) + ((LL)n-s[i])*(w[u]^w[i]);
        dfs2(i, u, n);
    }
}

void solve() {
    int n; cin >> n;
    for (int i = 0; i <= n+2; ++i) {
        vec[i].clear();
        w[i] = s[i] = f[i] = 0;
    }
    for (int i = 1; i <= n; ++i) cin >> w[i];
    for (int i = 1; i <= n-1; ++i) {
        int u, v; cin >> u >> v;
        vec[u].push_back(v), vec[v].push_back(u);
    }

    dfs1(1, -1);
    dfs2(1, -1, n);

    for (int i = 1; i <= n; ++i) cout << f[i] << " ";
    cout << endl;
}

int main()
{
    int T; cin >> T;

```

```
while (T -- ) { solve(); }  
return 0;  
}
```

## P2986 [USACO10MAR] Great Cow Gathering G

这个题目是第一题的变形题，跟第一题思路基本一致，与第一题的不同处是此题的每个点上有不同的点权

此题不提供具体代码，要求同学们课下独立完成