

树形dp练习3

人员

周子航、高可维、杨青悠、于迦浩、蔡云翔、石宇焮、李佳声、王崇宇、孙乐涵、窦浩轩 到课

作业

<https://www.luogu.com.cn/contest/158345>

B(没有上司的舞会)、C(Cut 'em all)两道题要求10分钟以内写出来

<https://www.luogu.com.cn/contest/160423>

邀请码: 97dv

A、B题必做，C题选做

课堂表现

周子航、石宇焮 2位同学在课下完成了选做题目，提出表扬!!!

课堂整体纪律比较好，于迦浩、窦浩轩、孙乐涵 3位同学上课听讲以及回答问题比较积极，提出表扬!!

课堂内容

P4084 [USACO17DEC] Barn Painting G

$f[u][k]$: 以u为根的子树中，当u节点的颜色为k时的方案数 ($1 \leq k \leq 3$)

当u的孩子节点有v1, v2, v3共3个子节点时

u一开始没有color, $f[u][1] = f[u][2] = f[u][3] = 1$

反之, $f[u][\text{color}[u]] = 1$

$f[u][1] = (f[v1][2] + f[v1][3]) * (f[v2][2] + f[v2][3]) * (f[v3][2] + f[v3][3])$

$f[u][2] = (f[v1][1] + f[v1][3]) * (f[v2][1] + f[v2][3]) * (f[v3][1] + f[v3][3])$

$f[u][3] = (f[v1][1] + f[v1][2]) * (f[v2][1] + f[v2][2]) * (f[v3][1] + f[v3][2])$

最后答案为 $f[\text{root}][1] + f[\text{root}][2] + f[\text{root}][3]$

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 1e5 + 5;
const int mod = 1e9 + 7;
vector<int> vec[maxn];
int f[maxn][4], w[maxn];

void dfs(int u, int fa) {
    if (w[u]) f[u][w[u]] = 1;
    else f[u][1] = f[u][2] = f[u][3] = 1;

    for (int i : vec[u]) {
        if (i == fa) continue;
        dfs(i, u);
        for (int j = 1; j <= 3; ++j) {
            int res = 0;
            for (int k = 1; k <= 3; ++k) {
                if (j == k) continue;
                res = (res + f[i][k]) % mod;
            }
            f[u][j] = (LL)f[u][j] * res % mod;
        }
    }
}

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

    dfs(1, -1);

    int res = 0;
    for (int i = 1; i <= 3; ++i) res = (res + f[1][i]) % mod;
    cout << res << endl;
    return 0;
}

```

The Fair Nut and the Best Path

与 <U283565 树的直径> 很类似的一道题

$f[u]$: 以 u 为根节点时, 往下的子树中最大的单链权值和

状态转移:

设 u 有4个孩子节点 v_1, v_2, v_3, v_4 ; 4个边权分别为 d_1, d_2, d_3, d_4

$f[v_2] - d_2 \geq f[v_1] - d_1 \geq f[v_4] - d_4 \geq f[v_3] - d_3$

则 $f[u] = f[v_2] - d_2 + w[u]$

$ans = \max(ans, \max(0, f[v_2] - d_2) + \max(0, f[v_1] - d_1) + w[u])$

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 3e5 + 5;
int w[maxn];
struct node {
    int to, value;
};
vector<node> vec[maxn];
LL f[maxn];

void dfs(int u, int fa, LL& ans) {
    f[u] = w[u];
    vector<LL> vv;
    for (node it : vec[u]) {
        if (it.to == fa) continue;
        dfs(it.to, u, ans);
        vv.push_back(f[it.to] - it.value);
    }
    sort(vv.begin(), vv.end()); reverse(vv.begin(), vv.end());
    LL res = f[u];
    for (int i = 0; i < (int)vv.size() && i < 2; ++i) {
        if (vv[i] < 0) break;
        res += vv[i];
    }
    ans = max(ans, res);

    if (vv.size() >= 1 && vv[0] > 0) f[u] += vv[0];
}

int main()
{
    ios::sync_with_stdio(false);
    cin.tie(0);

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

    LL ans = 0;
    dfs(1, -1, ans);
    cout << ans << endl;
    return 0;
}

```

Distance in Tree

定义状态：

$f[u][k]$ ：以 u 节点为根，往下子树中长度为 k 的链的数量

设 u 节点有4个孩子节点 v_1, v_2, v_3, v_4 , 则：

$f[u][0] = 1$

$f[u][1] = f[v_1][0] + f[v_2][0] + f[v_3][0] + f[v_4][0]$

$f[u][2] = f[v_1][1] + f[v_2][1] + f[v_3][1] + f[v_4][1]$

...

$f[u][k] = f[v_1][k-1] + f[v_2][k-1] + f[v_3][k-1] + f[v_4][k-1]$

$ans += f[u][k]$ // 以 u 节点为根，往下子树中长度为 k 的链的数量为答案做贡献

// 同时，还要考虑以 u 节点作为中间节点，往下延伸两条链的情况

$ans += f[v_2][0] * f[v_1][k-2] + \dots + f[v_2][k-2] * f[v_1][0]$

$ans += f[v_3][0] * (f[v_1][k-2] + f[v_2][k-2]) + \dots + f[v_3][k-2] * (f[v_1][0] + f[v_2][0])$

以上过程可以通过前缀和 $O(n*k)$ 实现

此题为课下思考题，不提供具体代码