

# 树上背包练习

---

## 人员

---

左子毅、朱奕鸣、杨洋、刘子淇、赵清航、周子航、于迦浩 到课

## 作业检查

---

左子毅 完成

朱奕鸣 未完成

杨洋 完成

刘子淇 完成

赵清航 完成

周子航 完成

于迦浩 完成

## 作业

---

<https://www.luogu.com.cn/contest/164040>

A、B 两个题目必做，C 题选做

<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

---

课堂上同学们隔空交流问题比较多，大家还是要静下心来好好想题做题

## 课堂内容

---

P2014 [CTSC1997] 选课

$f[u][k]$ : 以  $u$  为根的子树中, 选择  $k$  门课程时最多可以得到多少学分  
设有  $v_1, v_2, v_3, v_4$  4个孩子, 可以转化为一个分组背包问题  
在 $v_1$ 为根的子树中可以选择  $0, 1, 2, \dots$  个孩子, 可以认为这些物品中只能选择一个物品  
 $v_2, v_3, v_4$  4个孩子同理

```
#include <bits/stdc++.h>

using namespace std;

const int maxn = 300 + 5;
vector<int> vec[maxn];
int w[maxn], f[maxn][maxn];

void dfs(int u, int m) {
    f[u][1] = w[u];
    for (int i : vec[u]) {
        dfs(i, m);
        for (int j = m; j >= 0; --j) {
            for (int k = 1; k < j; ++k) f[u][j] = max(f[u][j], f[u][j-k] + f[i][k]);
        }
    }
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n; ++i) {
        int x; cin >> x >> w[i];
        vec[x].push_back(i);
    }
    dfs(0, m+1);
    cout << f[0][m+1] << endl;
    return 0;
}
```

## CF1153D Serval and Rooted Tree

$f[u]$ : 以  $u$  为根的子树中会由几个叶子来决定其值

1. 当  $u$  点为叶子节点时,  $f[u] = 1$
2. 当  $u$  点不为叶子节点时, 设其有  $v_1, v_2, v_3, v_4$  4个孩子  
则当  $w[u] = 1$  时,  $f[u] = \min(f[v_1], f[v_2], f[v_3], f[v_4])$   
当  $w[u] = 0$  时,  $f[u] = f[v_1] + f[v_2] + f[v_3] + f[v_4]$

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e5 + 5;
vector<int> vec[maxn];
int w[maxn], f[maxn];

void dfs(int u) {
    if (vec[u].empty()) { f[u] = 1; return; }

    int sum = 0, minn = 1e9;
    for (int i : vec[u]) {
        dfs(i);
        sum += f[i];
        minn = min(minn, f[i]);
    }
    if (w[u]) f[u] = minn;
    else f[u] = sum;
}

int main()
{
    int n; cin >> n;
    for (int i = 1; i <= n; ++i) cin >> w[i];
    for (int i = 2; i <= n; ++i) {
        int x; cin >> x; vec[x].push_back(i);
    }

    int sum = 0;
    for (int i = 1; i <= n; ++i) {
        if (vec[i].empty()) ++sum;
    }
    dfs(1);
    cout << sum - f[1] + 1 << endl;
    return 0;
}

```

## CF1454E Number of Simple Paths

首先可以认为 任意两点间存在2条不同的路径，此时总不同的路径数量是  $n*(n-1)$

之后，当认为  $v_1, v_2, v_3, v_4, v_5$  5个点在一个环上时，  
5个点往下延伸的5棵子树中的点 两两之间只有一条路径

所以，当  $v_1$  往下的子树中点的数量为  $k$  时，答案需要减去  $k*(k-1)/2$   
其余的  $v_2, v_3, v_4, v_5$  点同理

```

#include <bits/stdc++.h>

using namespace std;

typedef long long LL;
const int maxn = 2e5 + 5;
vector<int> vec[maxn];
int deg[maxn];
bool vis[maxn];

void topSort(int n) {
    queue<int> q;
    for (int i = 1; i <= n; ++i) {
        if (deg[i] == 1) q.push(i);
    }

    while (!q.empty()) {
        int u = q.front(); q.pop();
        vis[u] = false;
        for (int i : vec[u]) {
            if (--deg[i] == 1) q.push(i);
        }
    }
}

int dfs(int u, int fa) {
    int res = 1;
    for (int i : vec[u]) {
        if (!vis[i] && i != fa) res += dfs(i, u);
    }
    return res;
}

void solve() {
    int n; cin >> n;
    for (int i = 0; i < n + 2; ++i) {
        vec[i].clear();
        deg[i] = 0;
        vis[i] = true;
    }

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

    LL res = (LL)n * (n - 1);
    for (int i = 1; i <= n; ++i) {

```

```
        if (vis[i]) {
            int t = dfs(i, -1);
            res -= (LL)t * (t - 1) / 2;
        }
    }
    cout << res << endl;
}

int main()
{
    int T; cin >> T;
    while (T -- ) {
        solve();
    }
    return 0;
}
```