

树上动态规划-综合

人员

石宇赫、王崇宇、窦浩轩 到课

李佳声、蔡云翔、王梓同 未到

作业

<https://www.luogu.com.cn/contest/166104>

A、B、C、D 4道题

课堂表现

今天进行了一个小测验，有的同学做的比较好，有些同学细节考虑不是很全面，课下需要多加练习。

课堂内容

CF1900C Anji's Binary Tree

$f[u]$: 以 u 为根时，走到叶子节点的最少改变次数

$\text{if } (s[u][0]) \ f[u] = f[s[u][0]] + (w[u] \neq 'L')$

$\text{if } (s[u][1]) \ f[u] = f[s[u][1]] + (w[u] \neq 'R')$

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3e5 + 5;
const int inf = 0x3f3f3f3f;
int tr[maxn][2], f[maxn];
char s[maxn];

void dfs(int u) {
    int l = tr[u][0], r = tr[u][1];
    if (!l && !r) return;

    f[u] = inf;
    if (l) {
        dfs(l);
        f[u] = f[l] + (s[u] != 'L');
    }
    if (r) {
        dfs(r);
        f[u] = min(f[u], f[r] + (s[u] != 'R'));
    }
}

void solve() {
    int n; cin >> n;
    for (int i = 0; i <= n+2; ++i) f[i] = 0;

    cin >> (s+1);
    for (int i = 1; i <= n; ++i) cin >> tr[i][0] >> tr[i][1];
    dfs(1);

    cout << f[1] << endl;
}

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

```

CF1324F Maximum White Subtree

换根dp

先以 1 为根做一遍dfs，维护 f数组，然后从 1 开始进行转移

f[u]: 以 u 为根时，所连子图中最大的 cnt1-cnt2 值

维护 f数组:

```
f[u] = (w[u]==1?1:-1);
```

```
for (int i : vec[u]) f[u] += max(f[i], 0);
```

转移:

假设父节点是 u，子节点是 i

```
if (f[i] > 0) f[i] += max(f[u]-f[i], 0);
```

```
else f[i] += max(f[u], 0);
```

```

#include <bits/stdc++.h>

using namespace std;

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

void dfs1(int u, int fa) {
    f[u] = (w[u]==1?-1);
    for (int i : vec[u]) {
        if (i == fa) continue;
        dfs1(i, u);
        f[u] += max(f[i], 0);
    }
}

void dfs2(int u, int fa) {
    for (int i : vec[u]) {
        if (i == fa) continue;
        if (f[i] > 0) f[i] += max(0, f[u]-f[i]);
        else f[i] += max(0, f[u]);
        dfs2(i, u);
    }
}

int main()
{
    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; cin >> u >> v;
        vec[u].push_back(v), vec[v].push_back(u);
    }

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

    for (int i = 1; i <= n; ++i) cout << f[i] << " ";
    cout << endl;
    return 0;
}

```

P1273 有线电视网

$f[u][k]$: 以 u 为根, 选 k 个叶子节点时, 所得费用 与 所需代价 的差值的最大值

```
for (int j = min(s[u], m); j >= 1; --j) {  
    for (int k = 1; k <= min(s[i], j); ++k) {  
        f[u][j] = max(f[u][j], f[u][j-k] + f[i][k] - v);  
    }  
}
```

```

#include <bits/stdc++.h>

using namespace std;

const int maxn = 3000 + 5;
const int inf = 0x3f3f3f3f;
struct edge {
    int to, value;
};
vector<edge> vec[maxn];
int w[maxn], f[maxn][maxn], s[maxn];

void dfs(int u, int n, int m) {
    if (u >= n-m+1) { f[u][1] = w[u]; s[u] = 1; return; }

    for (int j = m; j >= 1; --j) f[u][j] = -inf;
    for (edge it : vec[u]) {
        int i = it.to, v = it.value;
        dfs(i, n, m);
        s[u] += s[i];
        for (int j = min(s[u], m); j >= 1; --j) {
            for (int k = 1; k <= min(s[i], j); ++k) {
                f[u][j] = max(f[u][j], f[u][j-k] + f[i][k] - v);
            }
        }
    }
}

int main()
{
    int n, m; cin >> n >> m;
    for (int i = 1; i <= n-m; ++i) {
        int k; cin >> k;
        while (k -- ) {
            int to, value; cin >> to >> value;
            vec[i].push_back({to, value});
        }
    }
    for (int i = n-m+1; i <= n; ++i) cin >> w[i];
    dfs(1, n, m);
    for (int i = m; i >= 0; --i) {
        if (f[1][i] >= 0) {
            cout << i << endl;
            break;
        }
    }
    return 0;
}

```