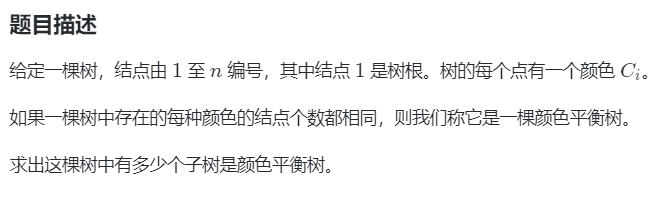
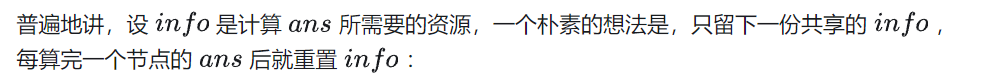
例题：[蓝桥杯：颜色平衡树](https://www.lanqiao.cn/problems/3504/learning/?page=1&first_category_id=1&second_category_id=3&status=3)





**void** **add**(**int** u) { */\* 加入u对info的影响 \*/* }

**void** **del**(**int** u) { */\* 清除u对info的影响 \*/* }

**void** **addsubtree**(**int** u, **int** fa **=** **-**1) {

add(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

addsubtree(v, u);

}

**void** **delsubtree**(**int** u, **int** fa **=** **-**1) {

del(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

delsubtree(v, u);

}

**void** **dfs**(**int** u, **int** fa **=** **-**1) {

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

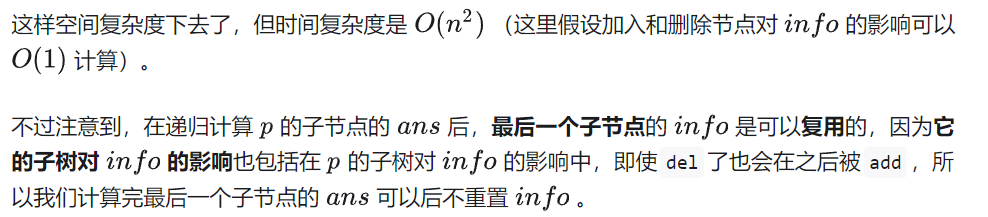
dfs(v, u);

addsubtree(u, fa);

*/\* 此处利用info计算ans \*/*

delsubtree(u, fa);

}



**void** **add**(**int** u) { */\* 加入u对info的影响 \*/* }

**void** **del**(**int** u) { */\* 清除u对info的影响 \*/* }

**void** **addsubtree**(**int** u, **int** fa **=** **-**1) {

add(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

addsubtree(v, u);

}

**void** **delsubtree**(**int** u, **int** fa **=** **-**1) {

del(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

delsubtree(v, u);

}

**void** **dfs**(**int** u, **int** fa **=** **-**1, **bool** keep **=** 1) {

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

dfs(v, u, v **==** edges[u].back()); *// 保留最后一个子节点的影响*

add(u);

**for** (**auto** v : edges[u])

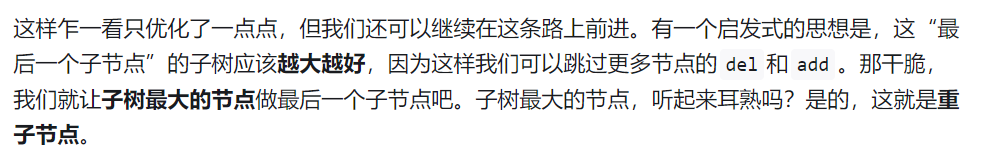
**if** (v **!=** fa **&&** v **!=** edges[u].back())

addsubtree(v, u); *// 最后一个子节点的子树的影响已经被保留了，所以不用再添加*

*/\* 此处利用info计算ans \*/*

**if** (**!**keep) delsubtree(u, fa);

}



**void** **add**(**int** u) { */\* 加入u对info的影响 \*/* }

**void** **del**(**int** u) { */\* 清除u对info的影响 \*/* }

**void** **addsubtree**(**int** u, **int** fa **=** **-**1) {

add(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

addsubtree(v, u);

}

**void** **delsubtree**(**int** u, **int** fa **=** **-**1) {

del(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa)

delsubtree(v, u);

}

**void** **dfs**(**int** u, **int** fa **=** **-**1, **bool** keep **=** 1) {

**for** (**auto** v : edges[u])

**if** (v **!=** fa **&&** v **!=** hson[u])

dfs(v, u, 0); *// 先遍历轻子节点*

**if** (hson[u]) dfs(hson[u], u, 1); *// 再遍历重子节点*

add(u);

**for** (**auto** v : edges[u])

**if** (v **!=** fa **&&** v **!=** hson[u]) *// 再遍历轻子节点*

addsubtree(v, u);

*/\* 此处利用info计算ans \*/*

**if** (**!**keep) delsubtree(u, fa);

}

