vector**<int>**sub[N];

**void** **dfs**(**int** u, **int** fa) {

id[u] **=** **++**tot;

**int** mx\_son **=** **-**1, mx\_sz **=** 0;

**for** (**int** v : g[u]) {

**if** (v **==** fa)**continue**;

dfs(v, u);

**if** (sub[id[v]].size() **>** mx\_sz) {

mx\_sz **=** sub[id[v]].size();

mx\_son **=** v;

}

}

**if** (mx\_son **!=** **-**1)id[u] **=** id[mx\_son];*//复制操作*

**for** (**int** v : g[u]) {

**if** (v **==** fa)**continue**;

**if** (v **==** mx\_son)**continue**;

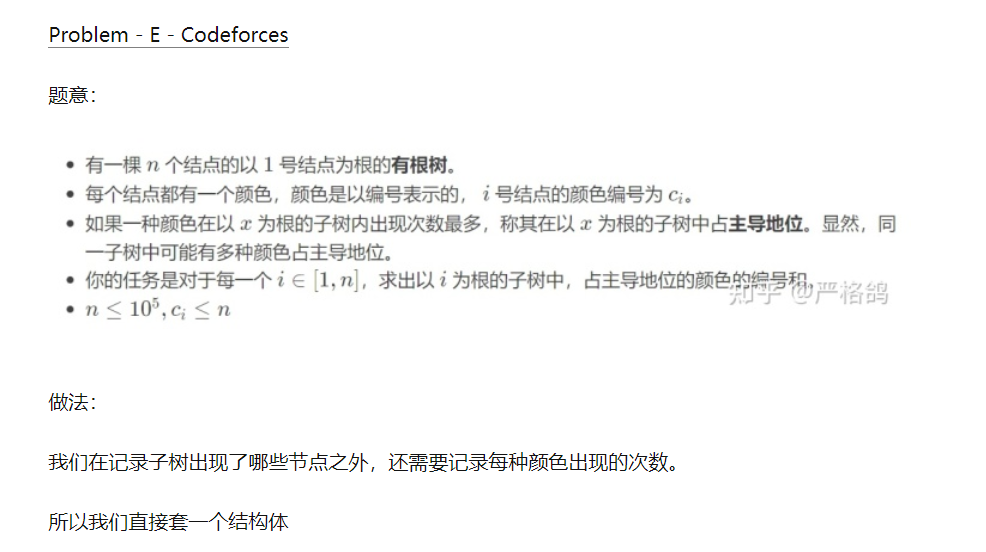
**for** (**int** son : sub[id[v]])

sub[id[u]].push\_back(son);

}

sub[id[u]].push\_back(u);

}

**const** **int** N **=** 1e5 **+** 5;

**int** n, c[N], id[N], tot **=** 0;

**struct** **node** {

**int** mx\_cnt **=** 0;*//最多的出现次数*

ll mx\_sum **=** 0;*//出现次数最多的颜色的编号和*

map**<int**, **int>**cnt;

vector**<int>**list;

**void** **add**(**int** u) {

cnt[c[u]]**++**;

**if** (cnt[c[u]] **>** mx\_cnt)mx\_cnt **=** cnt[c[u]], mx\_sum **=** c[u];

**else** **if** (cnt[c[u]] **==** mx\_cnt)mx\_sum **+=** c[u];

list.push\_back(u);

}

**int** **size**() { **return** list.size(); }

}sub[N];

ll ans[N];

vector**<int>**g[N];

**void** **dfs**(**int** u, **int** fa) {

id[u] **=** **++**tot;

**int** mx\_son **=** **-**1, mx\_sz **=** 0;

**for** (**int** v : g[u]) {

**if** (v **==** fa)**continue**;

dfs(v, u);

**if** (sub[id[v]].size() **>** mx\_sz) {

mx\_sz **=** sub[id[v]].size();

mx\_son **=** v;

}

}

**if**(mx\_son**!=-**1)id[u] **=** id[mx\_son];

**for** (**int** v : g[u]) {

**if** (v **==** fa)**continue**;

**if** (v **==** mx\_son)**continue**;

**for** (**int** son : sub[id[v]].list)

sub[id[u]].add(son);

}

sub[id[u]].add(u);

ans[u] **=** sub[id[u]].mx\_sum;

}

**void** **slove**() {

cin **>>** n;

**for** (**int** i **=** 1; i **<=** n; i**++**)cin **>>** c[i];

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

**int** u, v; cin **>>** u **>>** v;

g[u].push\_back(v);

g[v].push\_back(u);

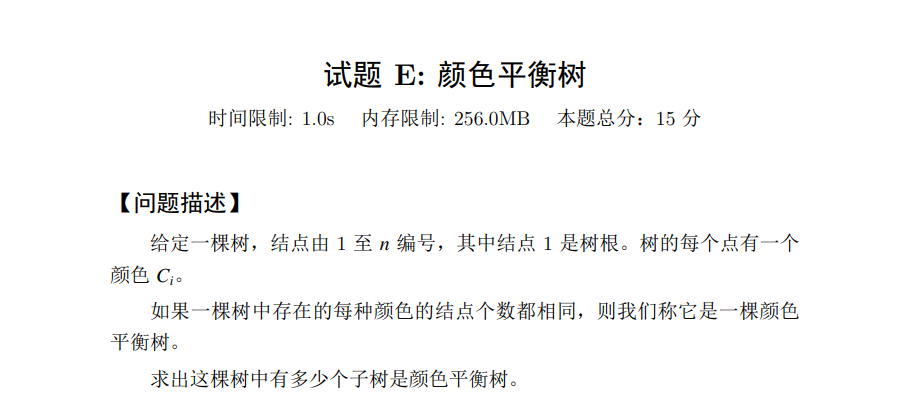
}

dfs(1, 0);

**for** (**int** i **=** 1; i **<=** n; i**++**)cout **<<** ans[i] **<<** " ";

cout **<<** endl;

}



也是一模一样的道理