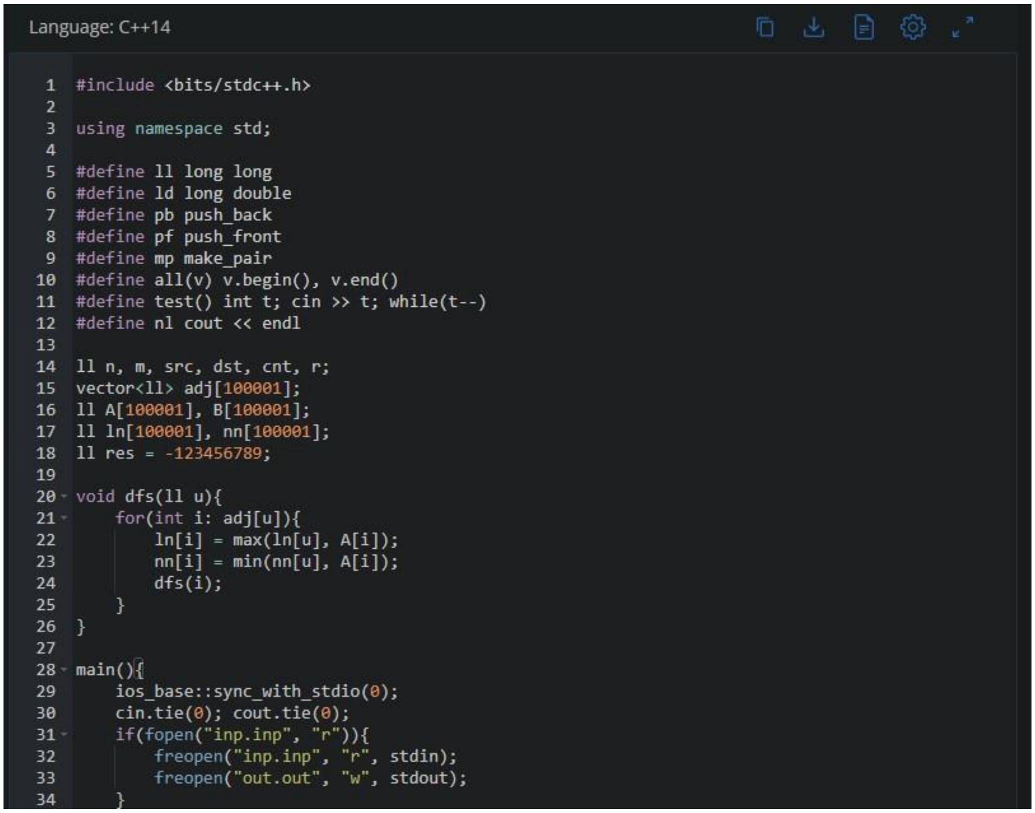
WORKSHEET 6

Student Name: Vivek Kumar **UID:** 21BCS8129

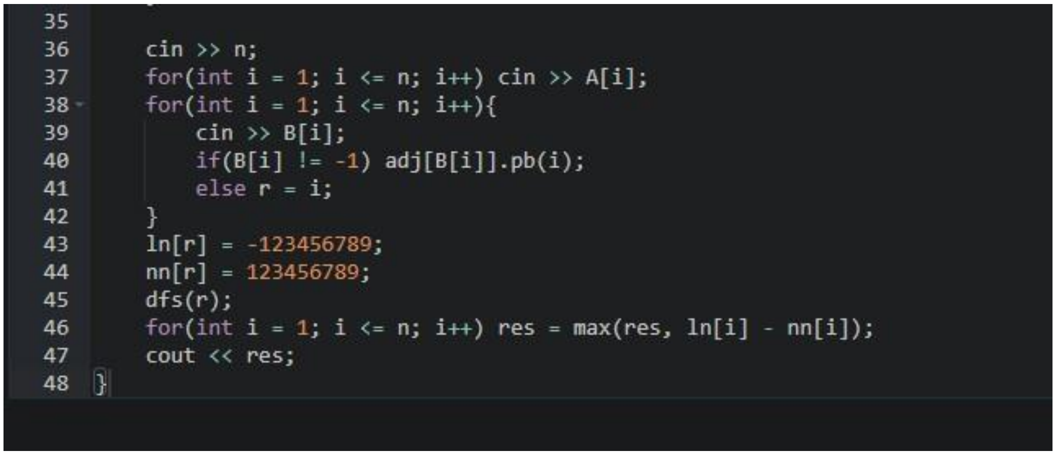
DOMAIN CAMP: 16-01-2023 to 28-01-2023 **Section/Group:** DWWC-77

Subject Name: IT Skills (DSA)

Question 1. FAMILY TREE

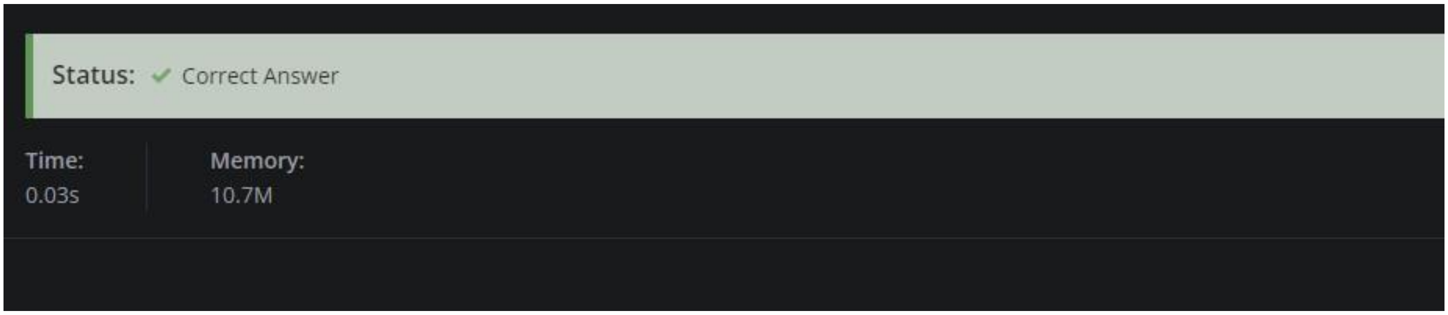


```cpp
#include <bits/stdc++.h>

using namespace std;

#define ll long long
#define ld long double
#define pb push_back
#define pf push_front
#define mp make_pair
#define all(v) v.begin(), v.end()
#define test() int t; cin >> t; while(t--)
#define nl cout << endl

ll n, m, src, dst, cnt, r;
vector<ll> adj[100001];
ll A[100001], B[100001];
ll ln[100001], nn[100001];
ll res = -123456789;

void dfs(ll u){
    for(int i: adj[u]){
        ln[i] = max(ln[u], A[i]);
        nn[i] = min(nn[u], A[i]);
        dfs(i);
    }
}

main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    if(fopen("inp.inp", "r")){
        freopen("inp.inp", "r", stdin);
        freopen("out.out", "w", stdout);
    }
```



```cpp

    cin >> n;
    for(int i = 1; i <= n; i++) cin >> A[i];
    for(int i = 1; i <= n; i++){
        cin >> B[i];
        if(B[i] != -1) adj[B[i]].pb(i);
        else r = i;
    }
    ln[r] = -123456789;
    nn[r] = 123456789;
    dfs(r);
    for(int i = 1; i <= n; i++) res = max(res, ln[i] - nn[i]);
    cout << res;
}
```

SOLUTION:

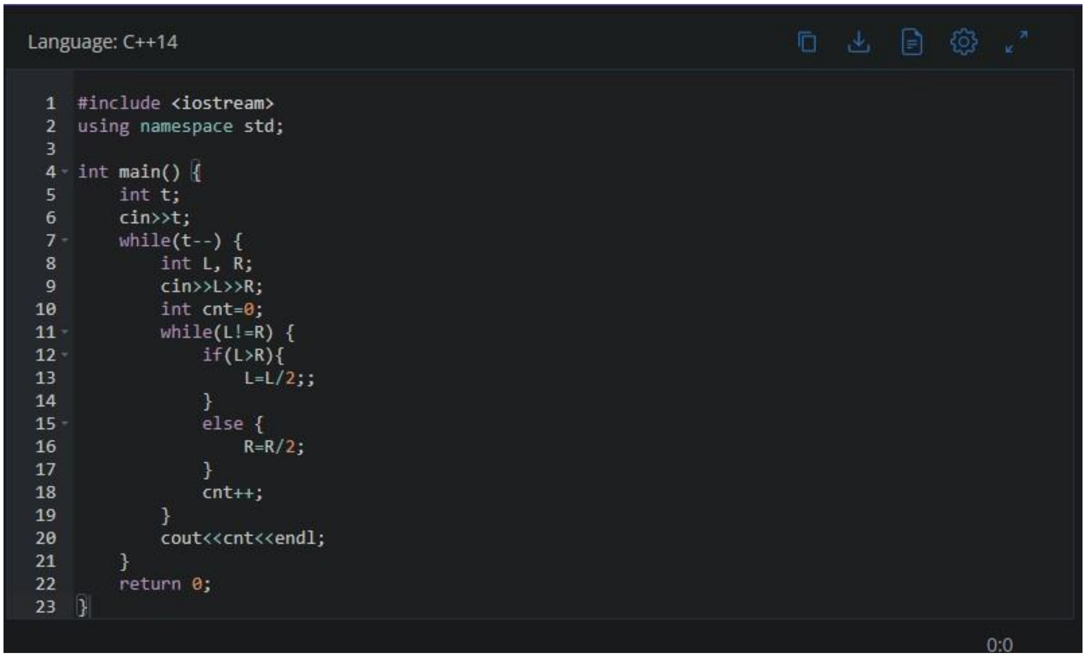


Status: ✓ Correct Answer

Time: 0.03s

Memory: 10.7M

Question 2. SHORTEST PATH IN BINARY TREES

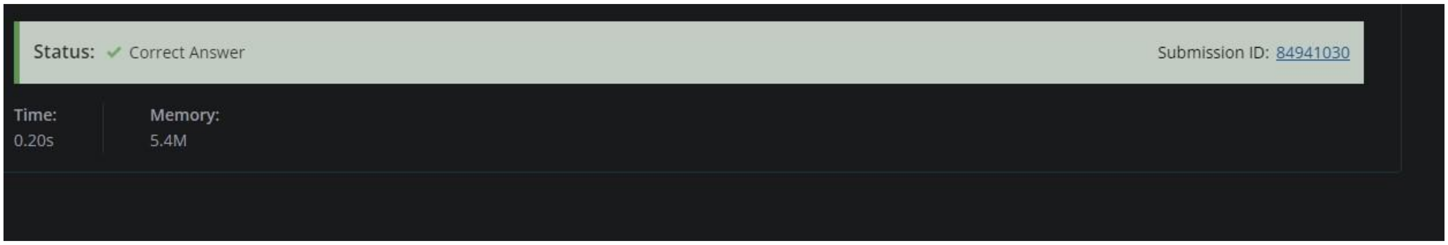


Language: C++14

```cpp
#include <iostream>
using namespace std;

int main() {
    int t;
    cin>>t;
    while(t--) {
        int L, R;
        cin>>L>>R;
        int cnt=0;
        while(L!=R) {
            if(L>R){
                L=L/2;;
            }
            else {
                R=R/2;
            }
            cnt++;
        }
        cout<<cnt<<endl;
    }
    return 0;
}
```

0:0

SOLUTION:



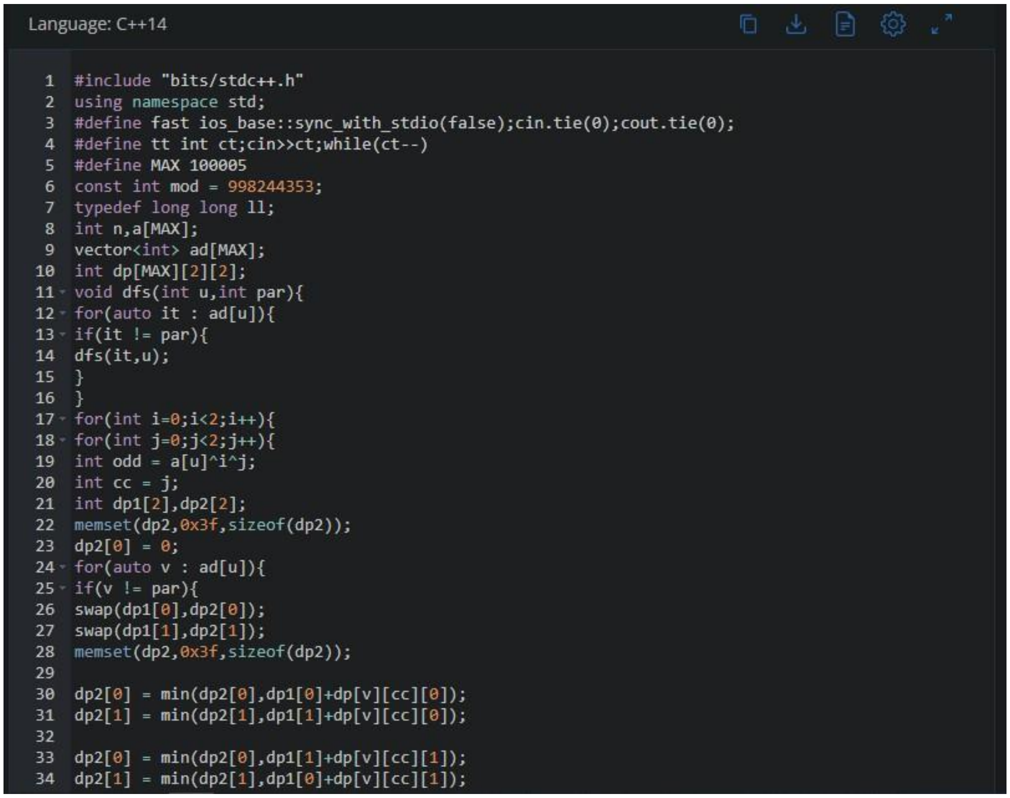
Status: ✓ Correct Answer

Submission ID: 84941030

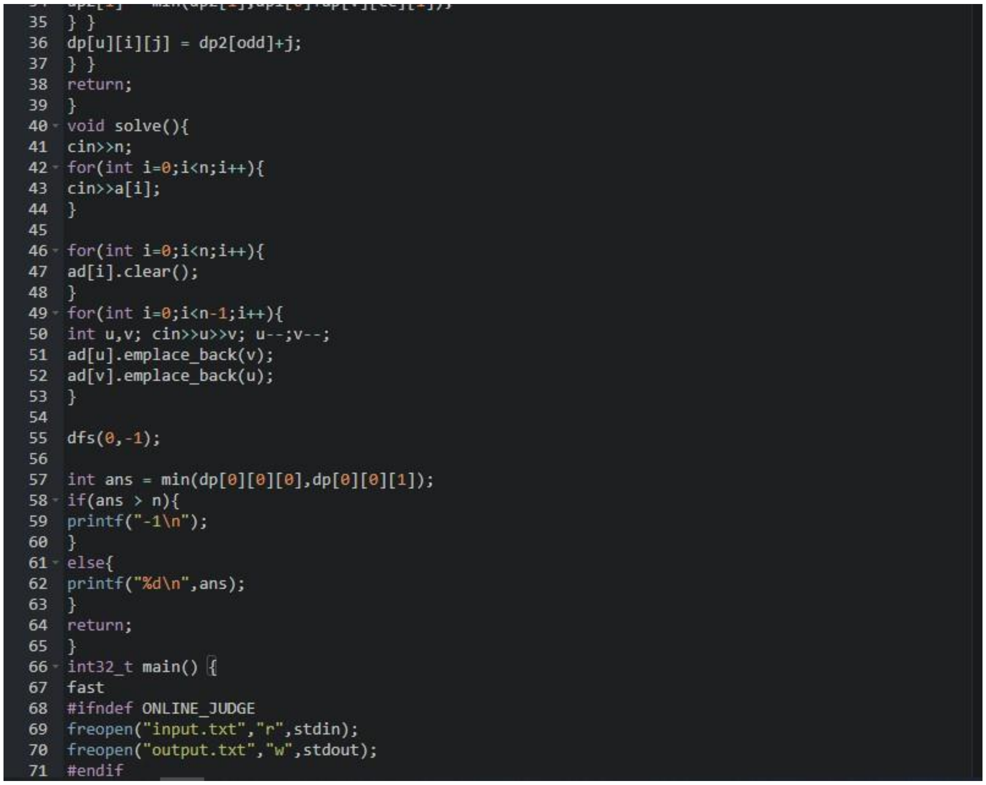
Time: 0.20s

Memory: 5.4M

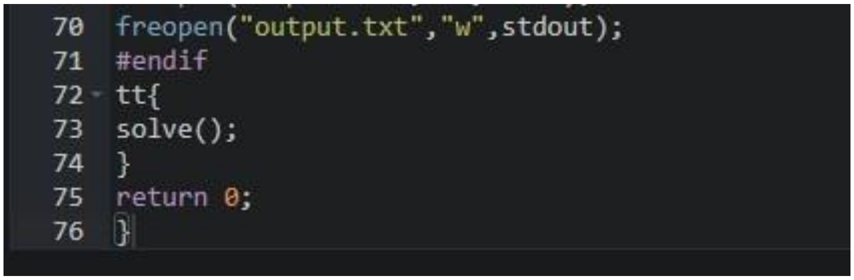
Question 3. BLACK AND WHITE TREE


Language: C++14

```cpp
#include "bits/stdc++.h"
using namespace std;
#define fast ios_base::sync_with_stdio(false);cin.tie(0);cout.tie(0);
#define tt int ct;cin>>ct;while(ct--)
#define MAX 100005
const int mod = 998244353;
typedef long long ll;
int n,a[MAX];
vector<int> ad[MAX];
int dp[MAX][2][2];
void dfs(int u,int par){
for(auto it : ad[u]){
if(it != par){
dfs(it,u);
}
}
for(int i=0;i<2;i++){
for(int j=0;j<2;j++){
int odd = a[u]^i^j;
int cc = j;
int dp1[2],dp2[2];
memset(dp2,0x3f,sizeof(dp2));
dp2[0] = 0;
for(auto v : ad[u]){
if(v != par){
swap(dp1[0],dp2[0]);
swap(dp1[1],dp2[1]);
memset(dp2,0x3f,sizeof(dp2));

dp2[0] = min(dp2[0],dp1[0]+dp[v][cc][0]);
dp2[1] = min(dp2[1],dp1[1]+dp[v][cc][0]);

dp2[0] = min(dp2[0],dp1[1]+dp[v][cc][1]);
dp2[1] = min(dp2[1],dp1[0]+dp[v][cc][1]);
```

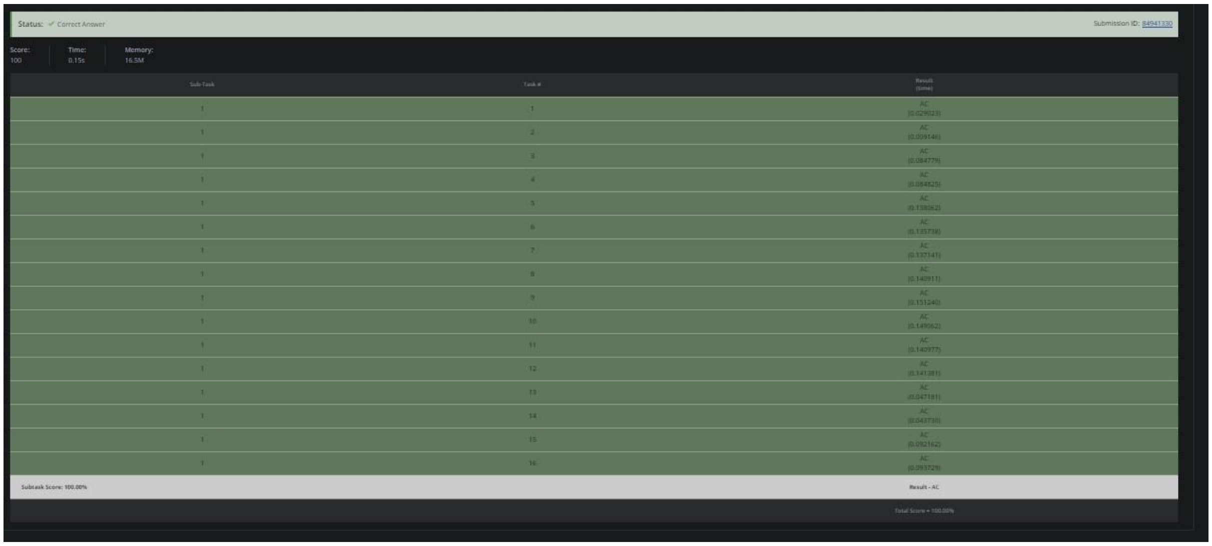


```cpp
} }
dp[u][i][j] = dp2[odd]+j;
} }
return;
}
void solve(){
cin>>n;
for(int i=0;i<n;i++){
cin>>a[i];
}

for(int i=0;i<n;i++){
ad[i].clear();
}
for(int i=0;i<n-1;i++){
int u,v; cin>>u>>v; u--;v--;
ad[u].emplace_back(v);
ad[v].emplace_back(u);
}

dfs(0,-1);

int ans = min(dp[0][0][0],dp[0][0][1]);
if(ans > n){
printf("-1\n");
}
else{
printf("%d\n",ans);
}
return;
}
int32_t main() {
fast
#ifndef ONLINE_JUDGE
freopen("input.txt","r",stdin);
freopen("output.txt","w",stdout);
#endif
```



```
freopen("output.txt","w",stdout);
#endif
tt{
solve();
}
return 0;
}
```

SOLUTION:



Status: ✓ Correct Answer

Submission ID: 84941330

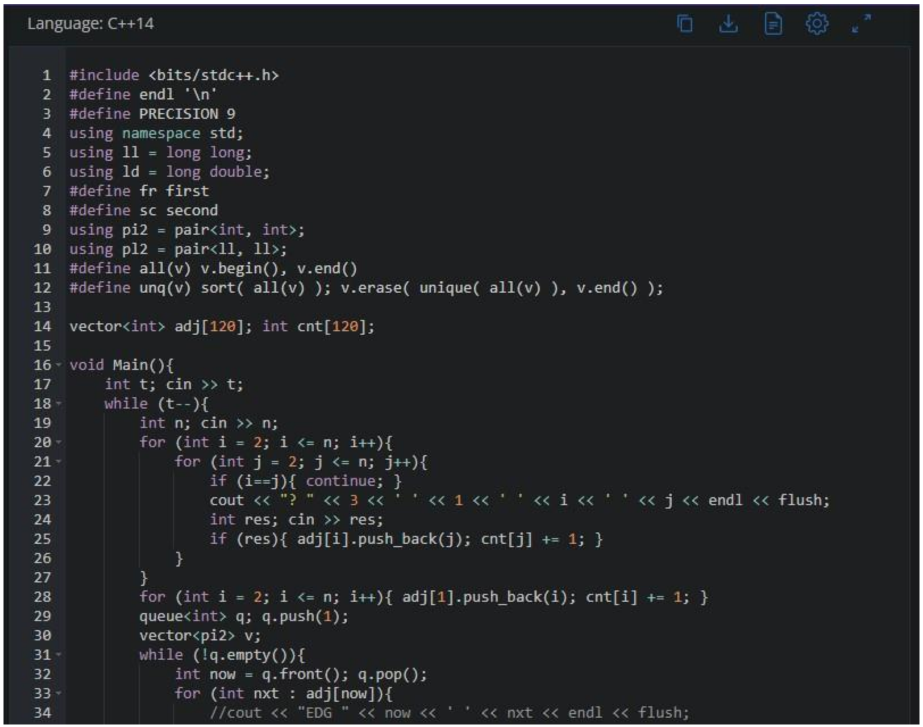
Score: 100

Time: 0.15s

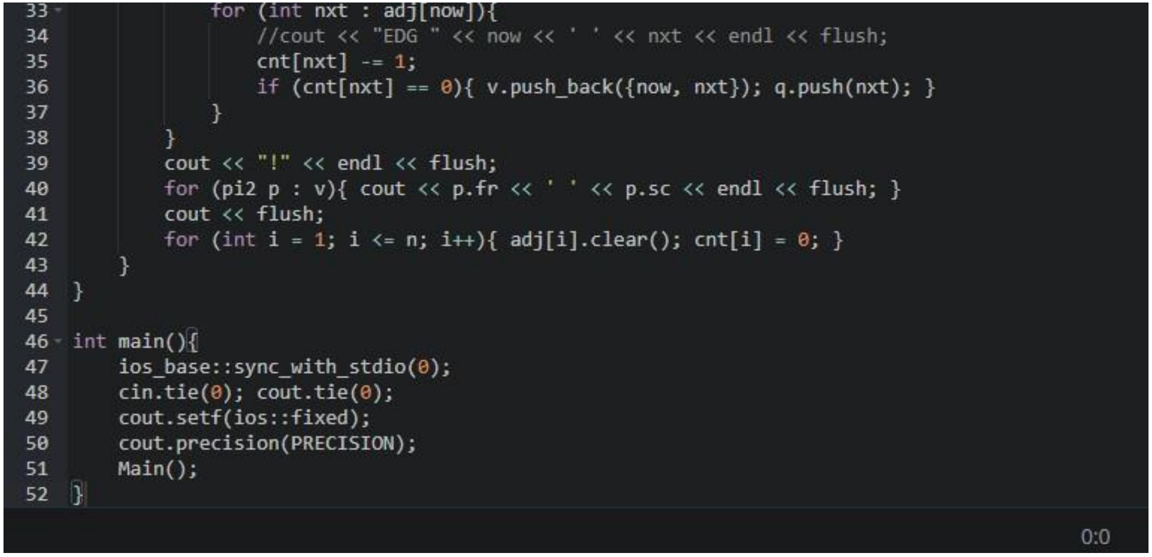
Memory: 16.5M

Sub-Task	Task #	Result (time)
1	1	AC (0.029023)
1	2	AC (0.009146)
1	3	AC (0.084779)
1	4	AC (0.084825)
1	5	AC (0.138062)
1	6	AC (0.135738)
1	7	AC (0.137141)
1	8	AC (0.140911)
1	9	AC (0.151240)
1	10	AC (0.149062)
1	11	AC (0.140977)
1	12	AC (0.141381)
1	13	AC (0.047181)
1	14	AC (0.043730)
1	15	AC (0.092162)
1	16	AC (0.093729)
Subtask Score: 100.00%		Result - AC
		Total Score = 100.00%

Question 4. SECRET TREE

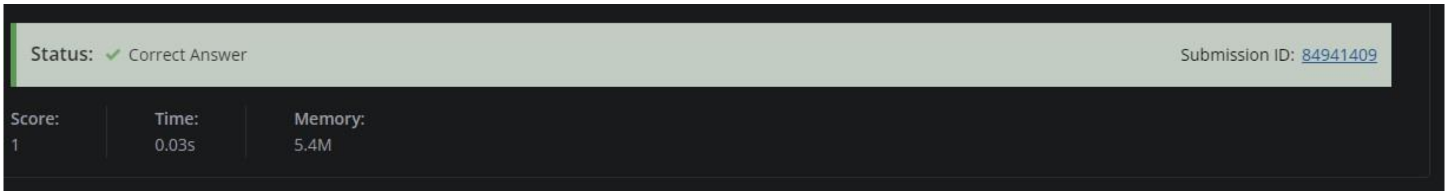


```cpp
#include <bits/stdc++.h>
#define endl '\n'
#define PRECISION 9
using namespace std;
using ll = long long;
using ld = long double;
#define fr first
#define sc second
using pi2 = pair<int, int>;
using pl2 = pair<ll, ll>;
#define all(v) v.begin(), v.end()
#define unq(v) sort( all(v) ); v.erase( unique( all(v) ), v.end() );

vector<int> adj[120]; int cnt[120];

void Main(){
    int t; cin >> t;
    while (t--){
        int n; cin >> n;
        for (int i = 2; i <= n; i++){
            for (int j = 2; j <= n; j++){
                if (i==j){ continue; }
                cout << "? " << 3 << ' ' << 1 << ' ' << i << ' ' << j << endl << flush;
                int res; cin >> res;
                if (res){ adj[i].push_back(j); cnt[j] += 1; }
            }
        }
        for (int i = 2; i <= n; i++){ adj[1].push_back(i); cnt[i] += 1; }
        queue<int> q; q.push(1);
        vector<pi2> v;
        while (!q.empty()){
            int now = q.front(); q.pop();
            for (int nxt : adj[now]){
                //cout << "EDG " << now << ' ' << nxt << endl << flush;
```



```cpp
            for (int nxt : adj[now]){
                //cout << "EDG " << now << ' ' << nxt << endl << flush;
                cnt[nxt] -= 1;
                if (cnt[nxt] == 0){ v.push_back({now, nxt}); q.push(nxt); }
            }
        }
        cout << "!" << endl << flush;
        for (pi2 p : v){ cout << p.fr << ' ' << p.sc << endl << flush; }
        cout << flush;
        for (int i = 1; i <= n; i++){ adj[i].clear(); cnt[i] = 0; }
    }
}

int main(){
    ios_base::sync_with_stdio(0);
    cin.tie(0); cout.tie(0);
    cout.setf(ios::fixed);
    cout.precision(PRECISION);
    Main();
}
```

SOLUTION:



Status: ✓ Correct Answer

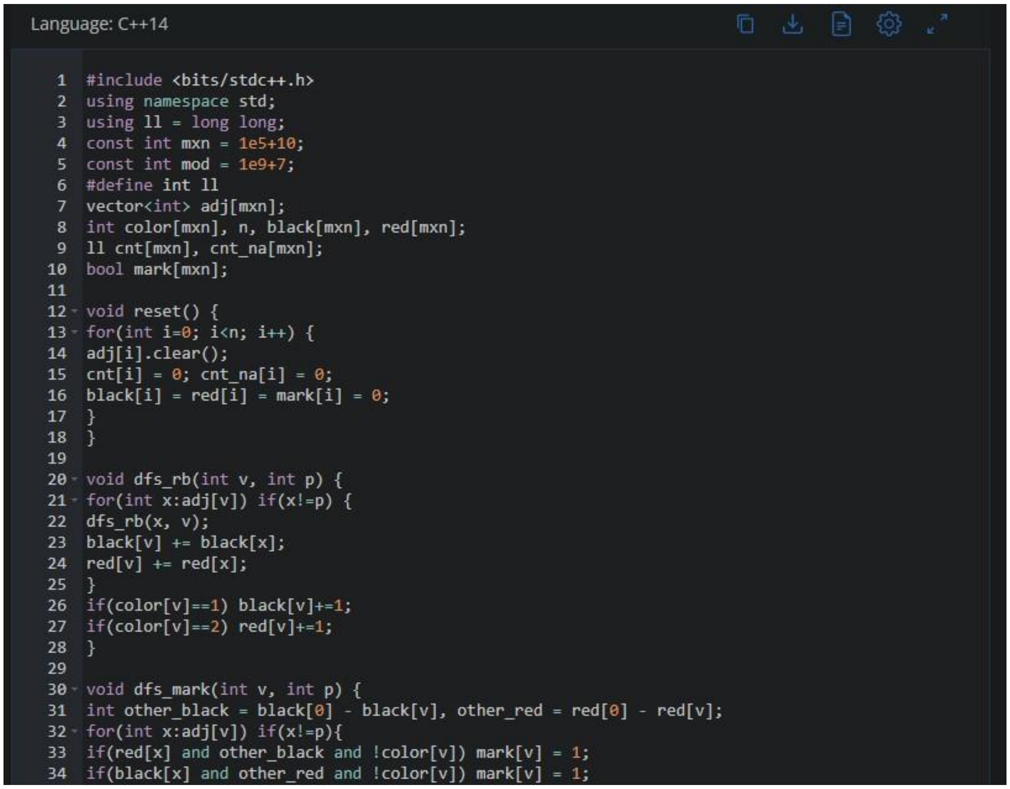
Submission ID: 84941409

Score:
1

Time:
0.03s

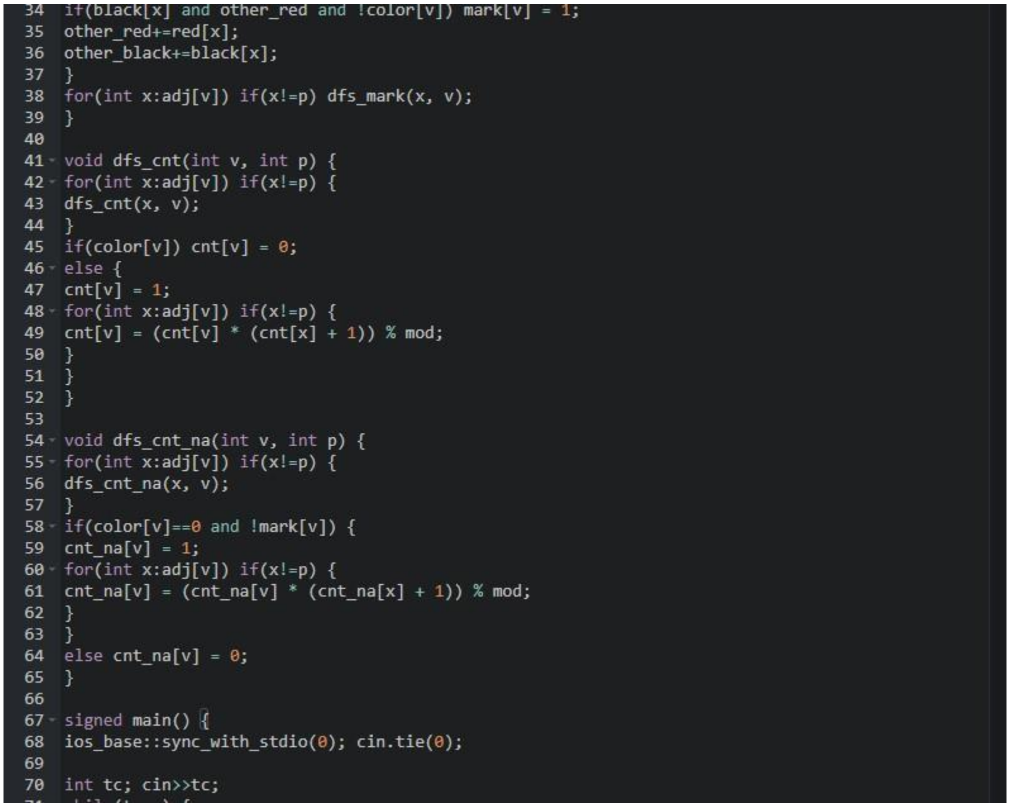
Memory:
5.4M

Question 5. BLACK AND RED VERTICES OF TREE

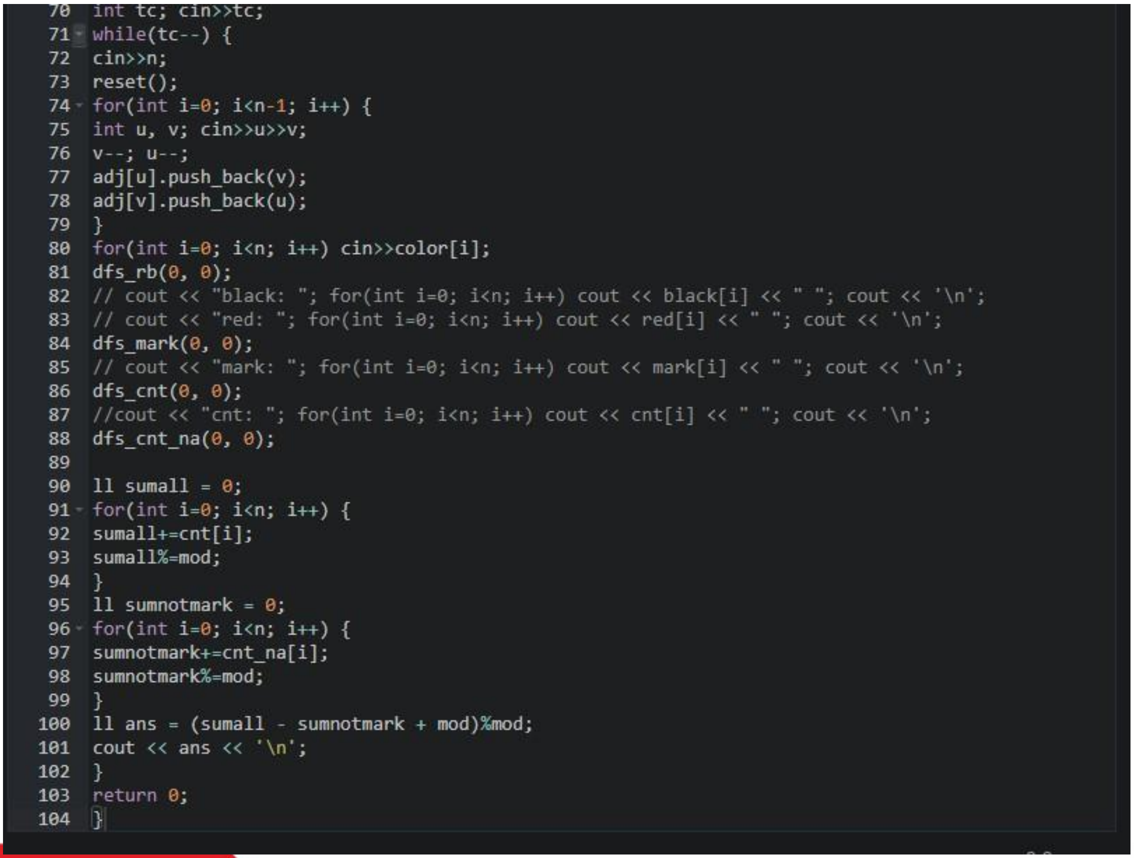


Language: C++14

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;
const int mxn = 1e5+10;
const int mod = 1e9+7;
#define int ll
vector<int> adj[mxn];
int color[mxn], n, black[mxn], red[mxn];
ll cnt[mxn], cnt_na[mxn];
bool mark[mxn];

void reset() {
for(int i=0; i<n; i++) {
adj[i].clear();
cnt[i] = 0; cnt_na[i] = 0;
black[i] = red[i] = mark[i] = 0;
}
}

void dfs_rb(int v, int p) {
for(int x:adj[v]) if(x!=p) {
dfs_rb(x, v);
black[v] += black[x];
red[v] += red[x];
}
if(color[v]==1) black[v]+=1;
if(color[v]==2) red[v]+=1;
}

void dfs_mark(int v, int p) {
int other_black = black[0] - black[v], other_red = red[0] - red[v];
for(int x:adj[v]) if(x!=p){
if(red[x] and other_black and !color[v]) mark[v] = 1;
if(black[x] and other_red and !color[v]) mark[v] = 1;
```



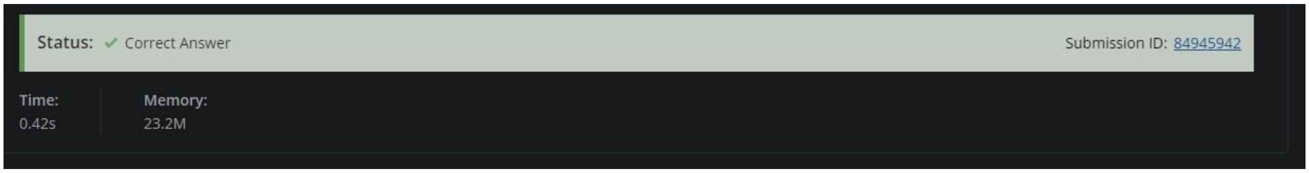
```
if(black[x] and other_red and !color[v]) mark[v] = 1;
other_red+=red[x];
other_black+=black[x];
}
for(int x:adj[v]) if(x!=p) dfs_mark(x, v);
}

void dfs_cnt(int v, int p) {
for(int x:adj[v]) if(x!=p) {
dfs_cnt(x, v);
}
if(color[v]) cnt[v] = 0;
else {
cnt[v] = 1;
for(int x:adj[v]) if(x!=p) {
cnt[v] = (cnt[v] * (cnt[x] + 1)) % mod;
}
}
}

void dfs_cnt_na(int v, int p) {
for(int x:adj[v]) if(x!=p) {
dfs_cnt_na(x, v);
}
if(color[v]==0 and !mark[v]) {
cnt_na[v] = 1;
for(int x:adj[v]) if(x!=p) {
cnt_na[v] = (cnt_na[v] * (cnt_na[x] + 1)) % mod;
}
}
else cnt_na[v] = 0;
}

signed main() {
ios_base::sync_with_stdio(0); cin.tie(0);

int tc; cin>>tc;
```



int tc; cin>>tc;

```
while(tc--) {
cin>>n;
reset();
for(int i=0; i<n-1; i++) {
int u, v; cin>>u>>v;
v--; u--;
adj[u].push_back(v);
adj[v].push_back(u);
}
for(int i=0; i<n; i++) cin>>color[i];
dfs_rb(0, 0);
// cout << "black: "; for(int i=0; i<n; i++) cout << black[i] << " "; cout << '\n';
// cout << "red: "; for(int i=0; i<n; i++) cout << red[i] << " "; cout << '\n';
dfs_mark(0, 0);
// cout << "mark: "; for(int i=0; i<n; i++) cout << mark[i] << " "; cout << '\n';
dfs_cnt(0, 0);
//cout << "cnt: "; for(int i=0; i<n; i++) cout << cnt[i] << " "; cout << '\n';
dfs_cnt_na(0, 0);

ll sumall = 0;
for(int i=0; i<n; i++) {
sumall+=cnt[i];
sumall%=mod;
}
ll sumnotmark = 0;
for(int i=0; i<n; i++) {
sumnotmark+=cnt_na[i];
sumnotmark%=mod;
}
ll ans = (sumall - sumnotmark + mod)%mod;
cout << ans << '\n';
}
return 0;
}
```

SOLUTION:

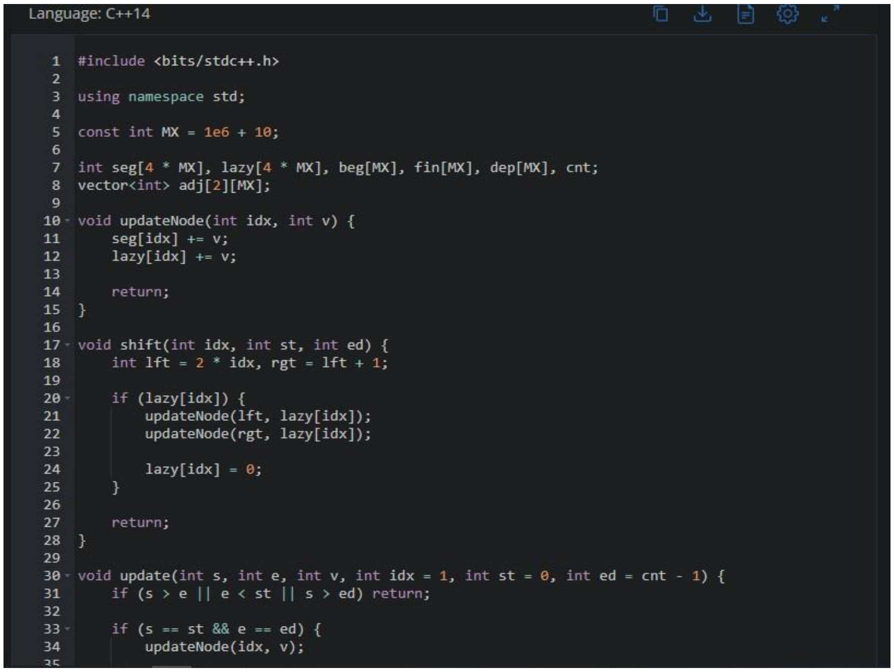


Status: ✓ Correct Answer Submission ID: 84945942

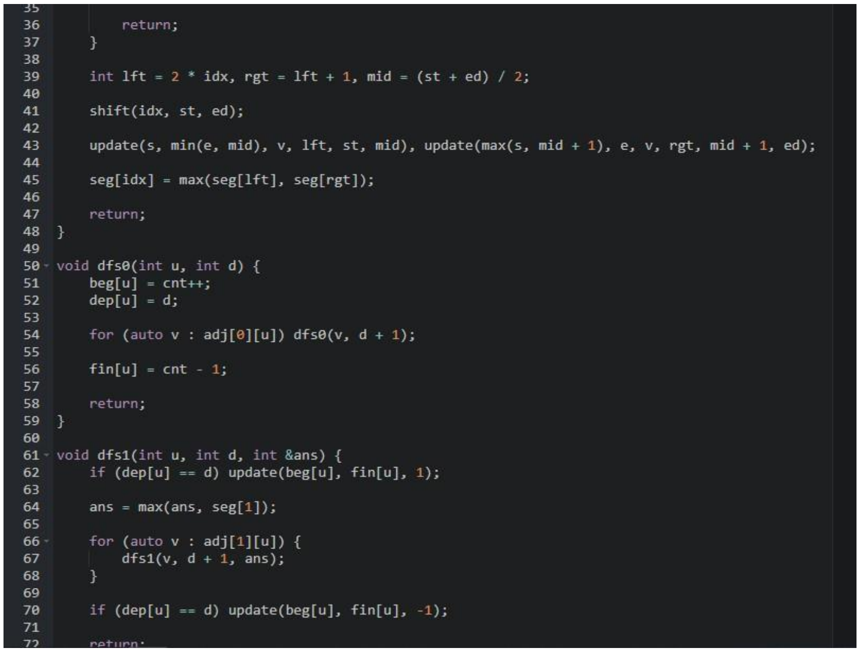
Time: 0.42s

Memory: 23.2M

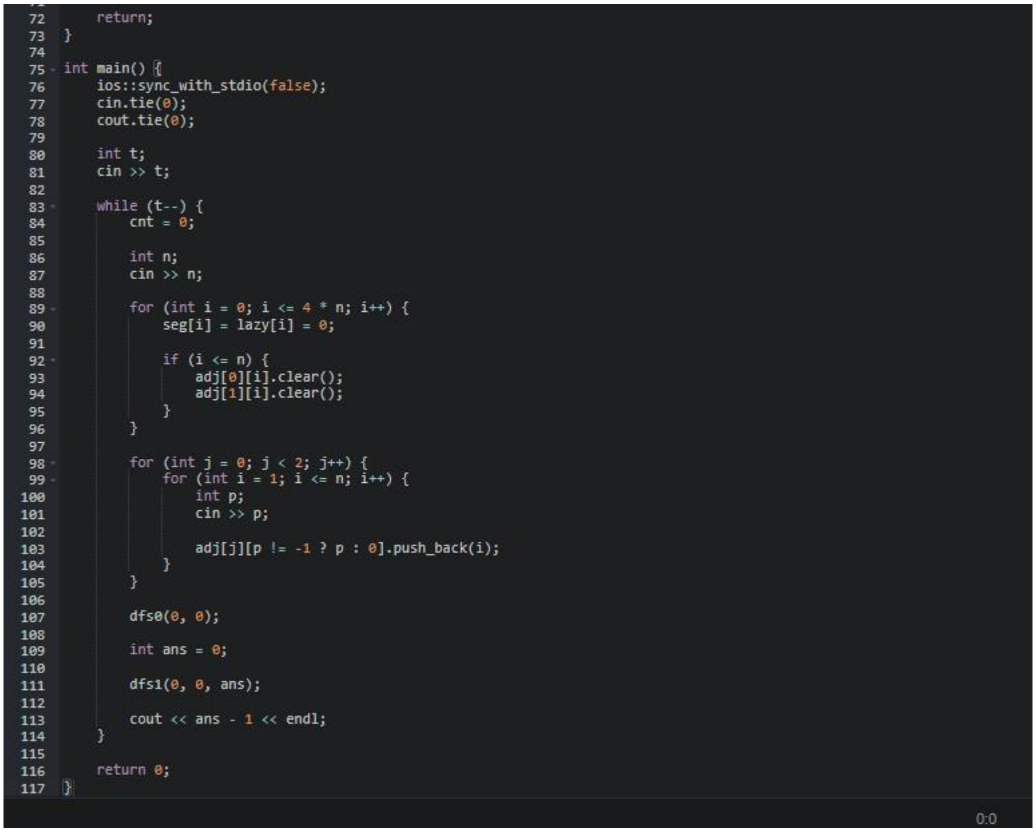
Question 6. COMMON ANCESTORS



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

using namespace std;

const int MX = 1e6 + 10;

int seg[4 * MX], lazy[4 * MX], beg[MX], fin[MX], dep[MX], cnt;
vector<int> adj[2][MX];

void updateNode(int idx, int v) {
    seg[idx] += v;
    lazy[idx] += v;

    return;
}

void shift(int idx, int st, int ed) {
    int lft = 2 * idx, rgt = lft + 1;

    if (lazy[idx]) {
        updateNode(lft, lazy[idx]);
        updateNode(rgt, lazy[idx]);

        lazy[idx] = 0;
    }

    return;
}

void update(int s, int e, int v, int idx = 1, int st = 0, int ed = cnt - 1) {
    if (s > e || e < st || s > ed) return;

    if (s == st && e == ed) {
        updateNode(idx, v);
```

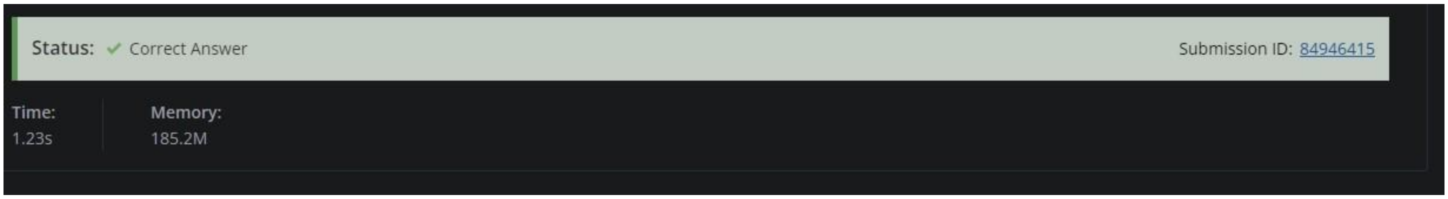


```

        return;
    }

    int lft = 2 * idx, rgt = lft + 1, mid = (st + ed) / 2;

    shift(idx, st, ed);

    update(s, min(e, mid), v, lft, st, mid), update(max(s, mid + 1), e, v, rgt, mid + 1, ed);

    seg[idx] = max(seg[lft], seg[rgt]);

    return;
}

void dfs0(int u, int d) {
    beg[u] = cnt++;
    dep[u] = d;

    for (auto v : adj[0][u]) dfs0(v, d + 1);

    fin[u] = cnt - 1;

    return;
}

void dfs1(int u, int d, int &ans) {
    if (dep[u] == d) update(beg[u], fin[u], 1);

    ans = max(ans, seg[1]);

    for (auto v : adj[1][u]) {
        dfs1(v, d + 1, ans);
    }

    if (dep[u] == d) update(beg[u], fin[u], -1);

    return;
```



```
    return;
}

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

    int t;
    cin >> t;

    while (t--) {
        cnt = 0;

        int n;
        cin >> n;

        for (int i = 0; i <= 4 * n; i++) {
            seg[i] = lazy[i] = 0;

            if (i <= n) {
                adj[0][i].clear();
                adj[1][i].clear();
            }
        }

        for (int j = 0; j < 2; j++) {
            for (int i = 1; i <= n; i++) {
                int p;
                cin >> p;

                adj[j][p != -1 ? p : 0].push_back(i);
            }
        }

        dfs0(0, 0);

        int ans = 0;

        dfs1(0, 0, ans);

        cout << ans - 1 << endl;
    }

    return 0;
}
```

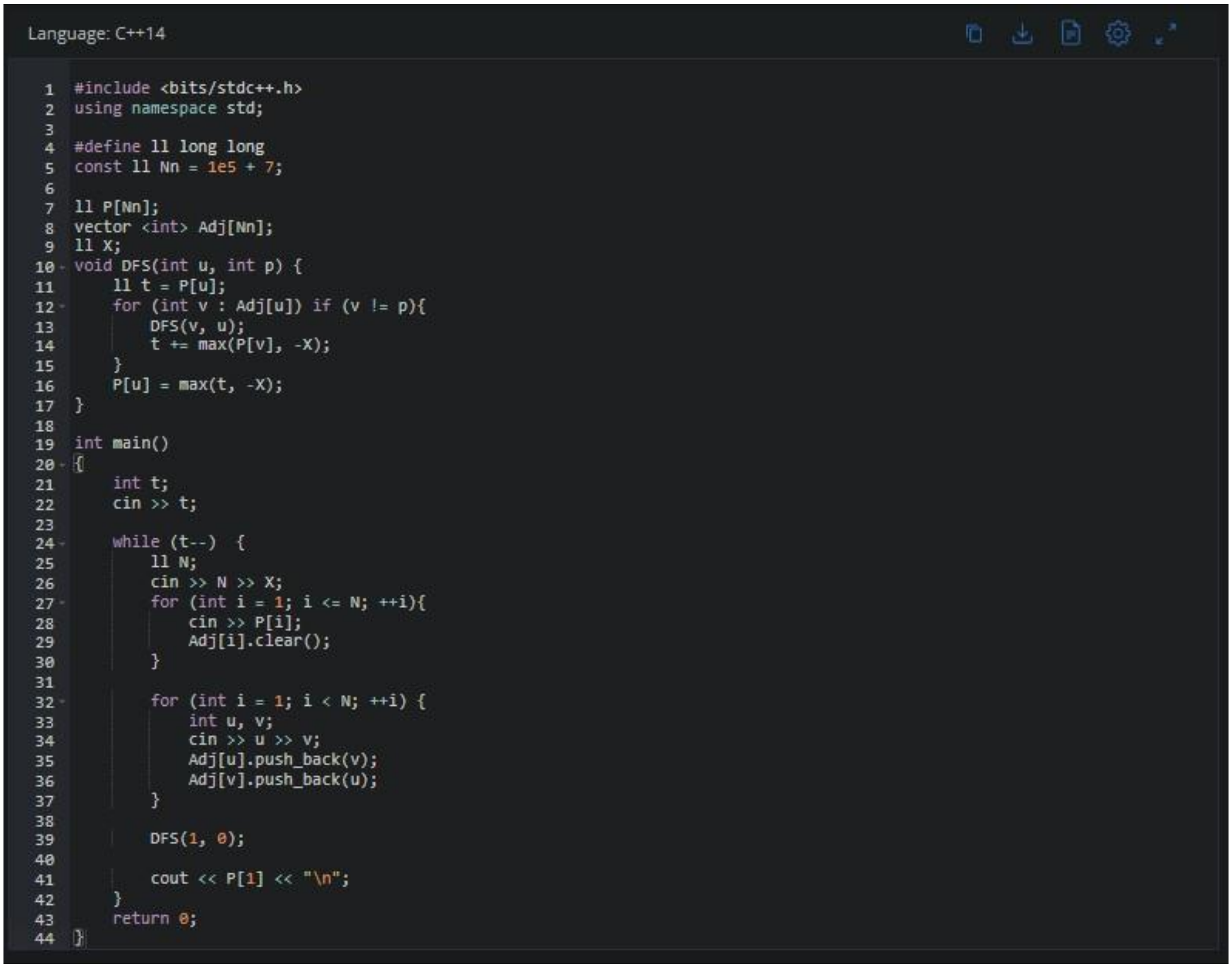
SOLUTION:



Status: ✓ Correct Answer   Submission ID: 84946415

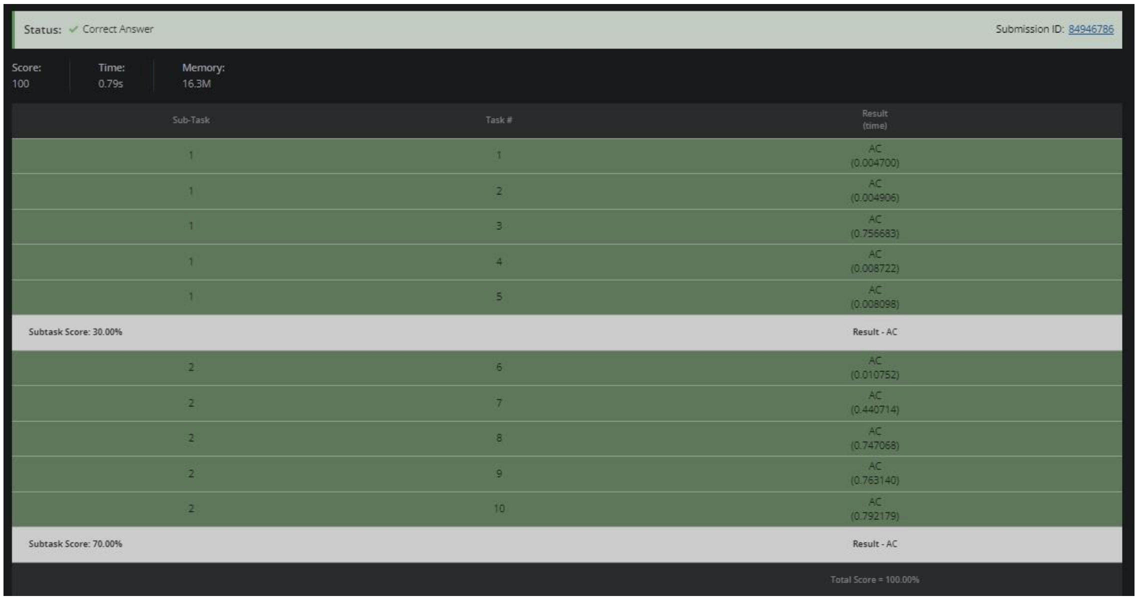
Time: 1.23s   Memory: 185.2M

Question 7. SUBTREE REMOVAL



```cpp
#include <bits/stdc++.h>
using namespace std;

#define ll long long
const ll Nn = 1e5 + 7;

ll P[Nn];
vector <int> Adj[Nn];
ll X;
void DFS(int u, int p) {
    ll t = P[u];
    for (int v : Adj[u]) if (v != p){
        DFS(v, u);
        t += max(P[v], -X);
    }
    P[u] = max(t, -X);
}

int main()
{
    int t;
    cin >> t;

    while (t--)  {
        ll N;
        cin >> N >> X;
        for (int i = 1; i <= N; ++i){
            cin >> P[i];
            Adj[i].clear();
        }

        for (int i = 1; i < N; ++i) {
            int u, v;
            cin >> u >> v;
            Adj[u].push_back(v);
            Adj[v].push_back(u);
        }

        DFS(1, 0);

        cout << P[1] << "\n";
    }
    return 0;
}
```

SOLUTION:

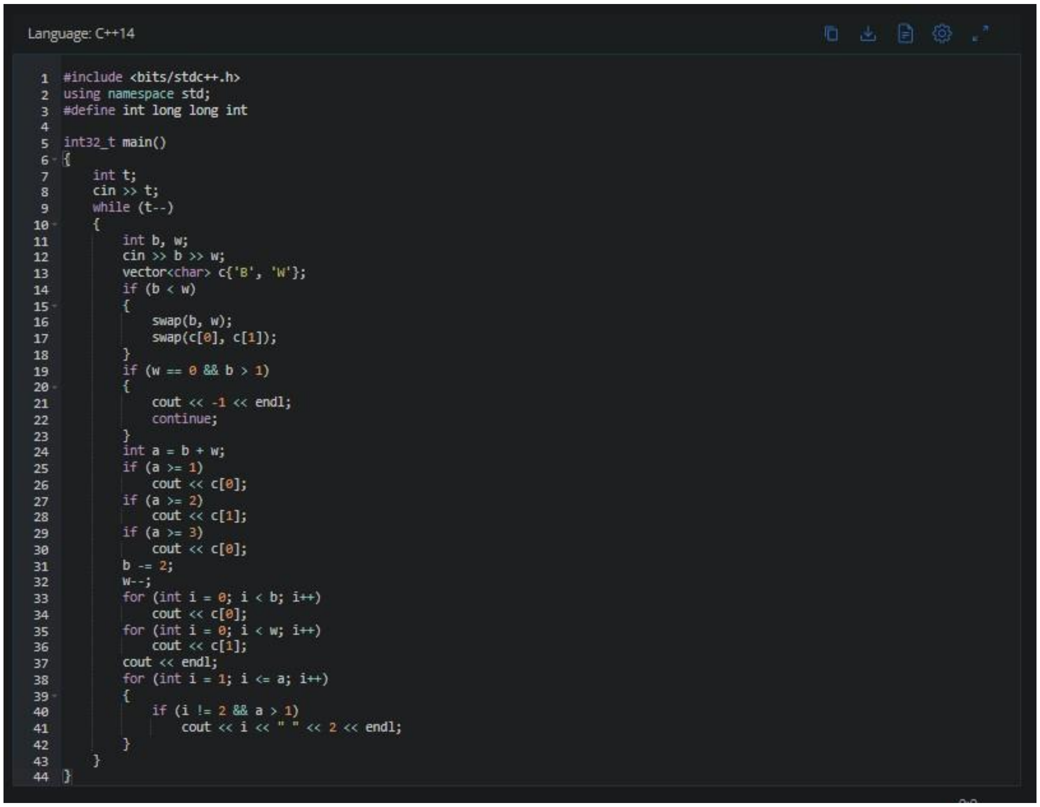


Status: ✓ Correct Answer Submission ID: 84946786

Score: 100 Time: 0.79s Memory: 16.3M

Sub-Task	Task #	Result (time)
1	1	AC (0.004700)
1	2	AC (0.004906)
1	3	AC (0.756683)
1	4	AC (0.008722)
1	5	AC (0.008098)
Subtask Score: 30.00%		Result - AC
2	6	AC (0.010752)
2	7	AC (0.440714)
2	8	AC (0.747068)
2	9	AC (0.763140)
2	10	AC (0.792179)
Subtask Score: 70.00%		Result - AC
		Total Score = 100.00%

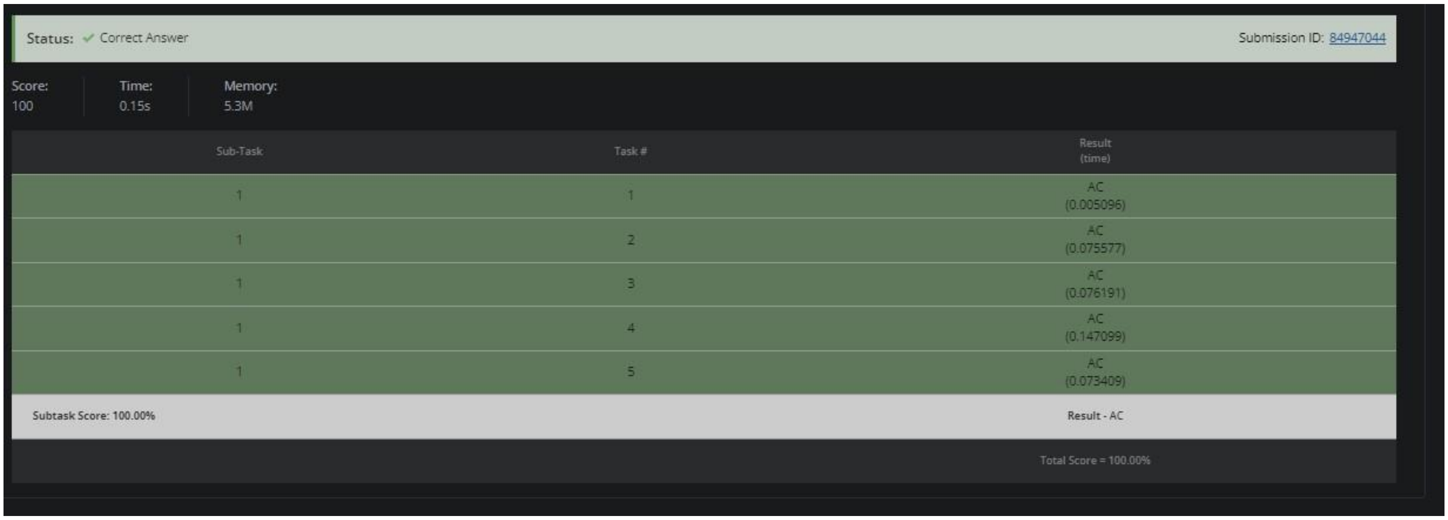
Question 8. ALTERNATING DIAMETER



Language: C++14

```cpp
#include <bits/stdc++.h>
using namespace std;
#define int long long int

int32_t main()
{
    int t;
    cin >> t;
    while (t--)
    {
        int b, w;
        cin >> b >> w;
        vector<char> c{'B', 'W'};
        if (b < w)
        {
            swap(b, w);
            swap(c[0], c[1]);
        }
        if (w == 0 && b > 1)
        {
            cout << -1 << endl;
            continue;
        }
        int a = b + w;
        if (a >= 1)
            cout << c[0];
        if (a >= 2)
            cout << c[1];
        if (a >= 3)
            cout << c[0];
        b -= 2;
        w--;
        for (int i = 0; i < b; i++)
            cout << c[0];
        for (int i = 0; i < w; i++)
            cout << c[1];
        cout << endl;
        for (int i = 1; i <= a; i++)
        {
            if (i != 2 && a > 1)
                cout << i << " " << 2 << endl;
        }
    }
}
```

SOLUTION:



Status: ✓ Correct Answer

Submission ID: 84947044

Score: 100

Time: 0.15s

Memory: 5.3M

Sub-Task	Task #	Result (time)
1	1	AC (0.005096)
1	2	AC (0.075577)
1	3	AC (0.076191)
1	4	AC (0.147099)
1	5	AC (0.073409)
Subtask Score: 100.00%		Result - AC
		Total Score = 100.00%