Worksheet 6

Domain Winter Winning Camp

Student Name: Abhinav Nath **UID:** 20BCS5562
Section: DWWC - 43 **Date of Performance:** 10th Jan, 2023
Subject Name: IT Skills

1. Shortest Path in Binary Trees

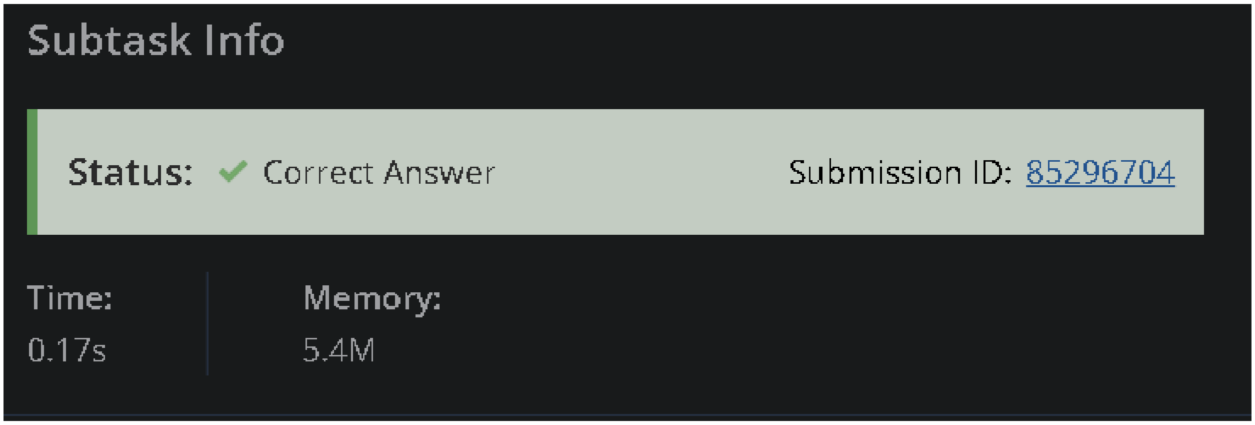
   Code:

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


#define scan(x,n) for(int i=0;i<(n);i++) cin>>(x)[i];

#define print(x,n) for(int i=0;i<(n);i++) cout<<(x)[i]<<" "; cout<<"\n";

#define all(x) (x).begin(), (x).end()

#define rall(x) (x).rbegin(), (x).rend()

#define ll long long


using namespace std;


class DSU{


public:

    int parent[100005];
int sz[100005];

DSU(int n){
    memset(parent,-1,sizeof(parent));
    memset(sz,1,sizeof(sz));
}

int find(int node){
    if (parent[node]==-1){
        return node;
    }
    return (parent[node]=find(parent[node]));
}

void uni(int node1,int node2){
    int a=find(node1);
    int b=find(node2);

    if (a!=b){
        if (sz[a]<sz[b]){
            parent[a]=b;
            }
            else if (sz[a]>sz[b]){
                parent[b]=a;
            }
            else{
                parent[b]=a;
                sz[a]++;
            }
        }
    }
};


int main() {

    ios_base::sync_with_stdio(false);
    cin.tie(NULL);

    int t;
    cin>>t;
```

```
while(t--)
{
    long long a,b;
    cin>>a>>b;

    int c=0;

    while(a!=b)
    {
        if(a>b)
        {
            a/=2;
            c++;
        }
        else
        {
            b/=2;
            c++;
        }
    }
```

```
        cout<<c<<endl;
    }



        return 0;
}
```

Output:



Subtask Info

Status: ✓ Correct Answer   Submission ID: 85296704

Time: 0.17s   Memory: 5.4M

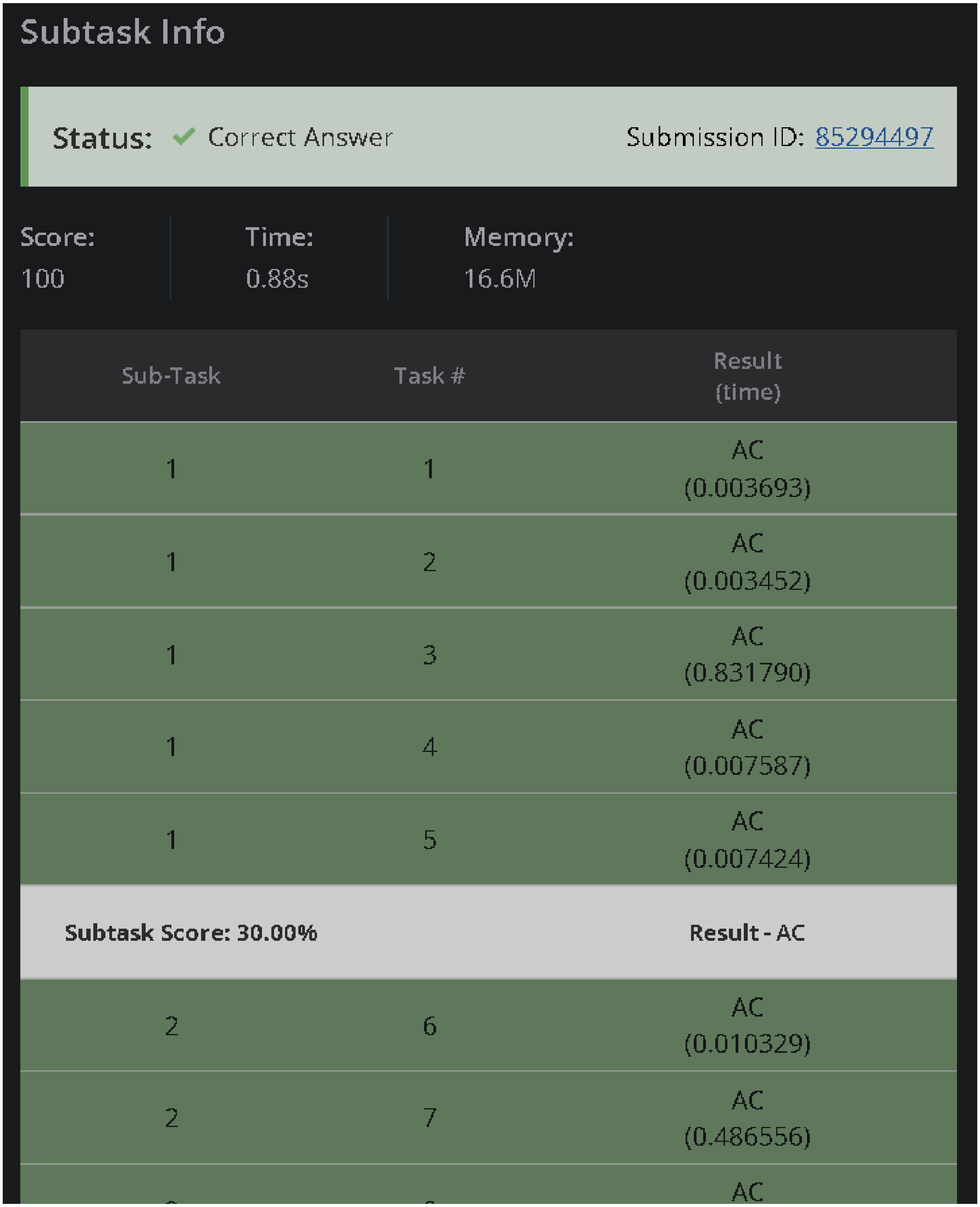
2. Subtree Removal

Code:

```
#include <iostream>

#include<bits/stdc++.h>

using namespace std;

vector<long> value;
```

```
long x;
long max(long a, long b)
{
    if (a >= b)
    {
        return a;
    }
    else
    {
        return b;
    }
}
long dfs(vector<long> v[], long a, long b)
{
    long i;
    long cal = value[a - 1];
    for (i = 0; i < v[a].size(); i++)
    {
        if (v[a][i] != b)
        {
            cal += dfs(v, v[a][i], a);
        }
    }
    return max(cal, -x);
}
int main()
{
    long t;
    cin >> t;
    while (t--)
    {
        value.clear();
        long n;
        cin >> n >> x;
        vector<long>
            v[n + 1];
        long i;
        long a, b;
        for (i = 1; i <= n; i++)
        {
            cin >> a;
            value.push_back(a);
        }
        for (i = 0; i < n - 1; i++)
        {
            cin >> a >> b;
            v[a].push_back(b);
            v[b].push_back(a);
        }
        long cal = dfs(v, 1, -1);
        printf("%ld\n", cal);
    }
}
```

Output:



Subtask Info

Status: ✓ Correct Answer Submission ID: 85294497

Score: 100 | Time: 0.88s | Memory: 16.6M

Sub-Task	Task #	Result (time)
1	1	AC (0.003693)
1	2	AC (0.003452)
1	3	AC (0.831790)
1	4	AC (0.007587)
1	5	AC (0.007424)
Subtask Score: 30.00%		**Result - AC**
2	6	AC (0.010329)
2	7	AC (0.486556)
		AC

3. Black and White Tree

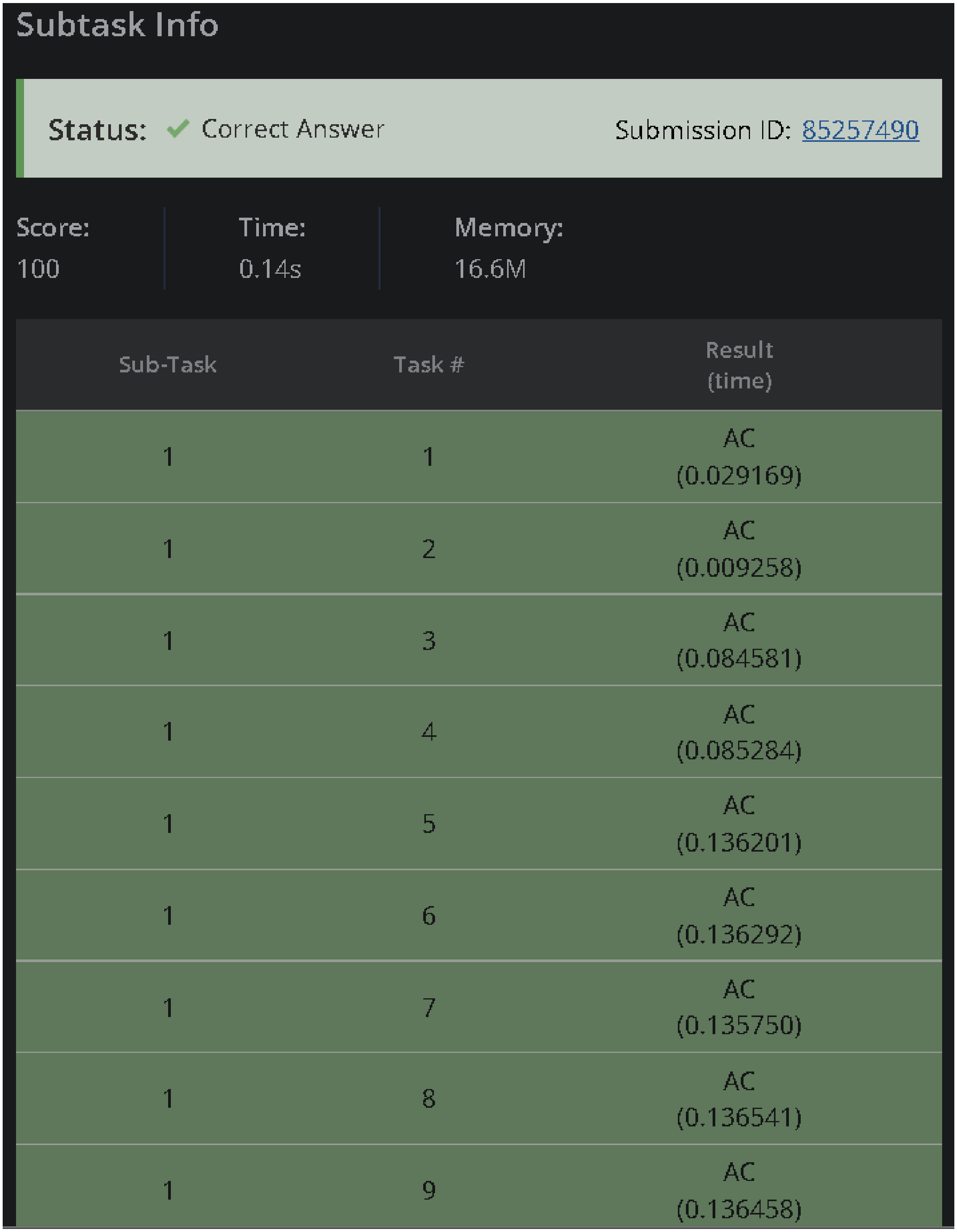
Code:

```
#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]);
} }
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
tt{
solve();
}
return 0;
}
```

Output:



Subtask Info

Status: ✓ Correct Answer   Submission ID: 85257490

Score: 100

Time: 0.14s

Memory: 16.6M

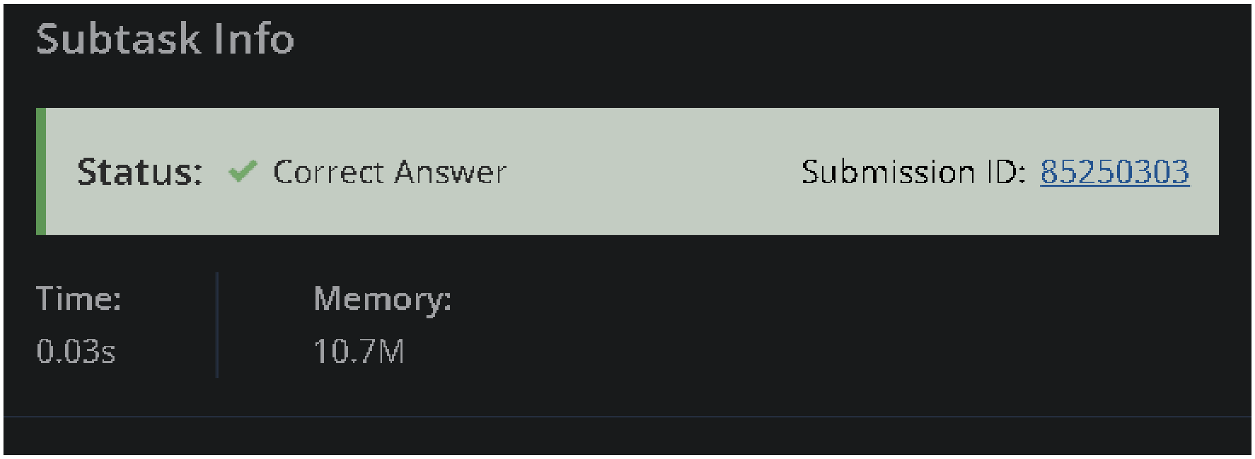
Sub-Task	Task #	Result (time)
1	1	AC (0.029169)
1	2	AC (0.009258)
1	3	AC (0.084581)
1	4	AC (0.085284)
1	5	AC (0.136201)
1	6	AC (0.136292)
1	7	AC (0.135750)
1	8	AC (0.136541)
1	9	AC (0.136458)

4. Family Tree

Code:

```
#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);
    }

    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;

}
```

Output:



Subtask Info

Status: ✓ Correct Answer

Submission ID: 85250303

Time:
0.03s

Memory:
10.7M

5. Common Ancestors

   Code:

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



using namespace std;



const int MX = 1e6 + 10;
```

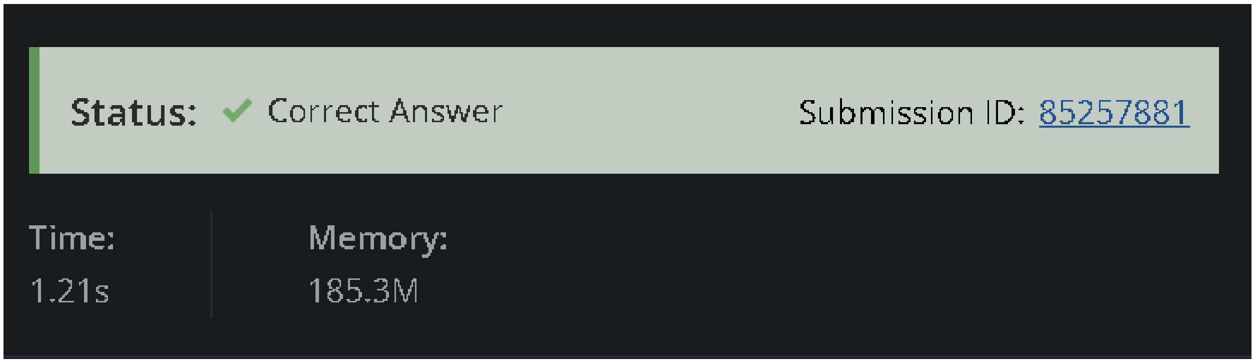
```
int seg[4 * MX], lazy[4 * MX], beg[MX], fin[MX], dep[MX], cnt;
vector<int> adj[2][MX];

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;
}

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;
}
```

Output:



Status: ✓ Correct Answer Submission ID: 85257881

Time: 1.21s

Memory: 185.3M

6. Lowest Common Ancestor

Code:

```
//#pragma gcc optimize("Ofast")

//#pragma GCC optimization("Ofast")

//#pragma optimize(Ofast)

#include <bits/stdc++.h>

#define MOD 1000000007
```

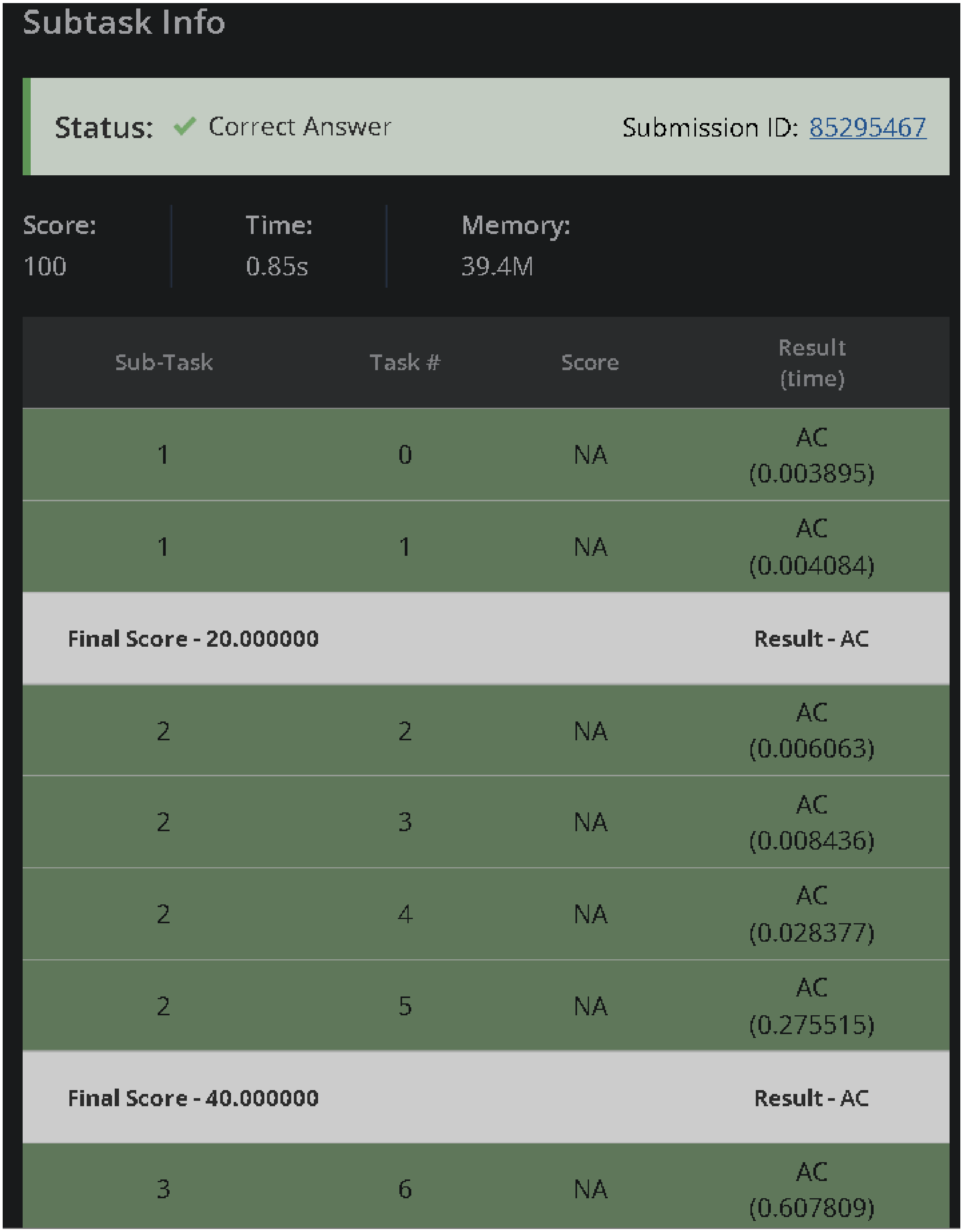
```
#define bugf cout << "Here" << endl;
#define bug(n) cout << (n) << endl;

using namespace std;

const long long INF = 2e18;
const int inf = 1e9 + 7;
string alphabet = "abcdefghijklmnopqrstuvwxyz";
const int base = 20;

//write variables
template <typename T>
ostream& operator<<(ostream& output, const vector<T> &arr){
    for(T a : arr) output << a << " ";
    return output;
}

//open file
void open(){
    if(fopen("input.inp", "r")){
        freopen("input.inp", "r", stdin);
        //freopen("output.out", "w", stdout);
    }
}

//pre init

//others struct or class

struct Ds{
    vector<int> ds;
    int len;
    Ds(int _len = 0){
        len = _len;
        ds.assign(len, -1);
    }
    void reinit(){
        ds.assign(len, -1);
    }
    int count(){
        int count = 0;
        for(int a : ds){
            if(a < 0) count++;
        }
        return count;
    }
    int find(int a){
        return ((ds[a] < 0) ? a : ds[a] = find(ds[a]));
    }
    void unionds(int a, int b){
        a = find(a);
        b = find(b);
        if(a > b) swap(a, b);
        if(a != b){
            ds[b] = a;
        }
    }
};

struct Point{
    double x, y;
    Point(){}
    Point(double _x, double _y) : x(_x), y(_y) {}
```

```
        double operator-(const Point &other){
                return (x - other.x)*(x - other.x) + (y - other.y)*(y - other.y);
        }

        bool operator>(const Point &other) const{
                if(x != other.x) return x > other.x;
                return y > other.y;
        }
};

template <class T>
struct Matrix{
        vector<vector<T>> matrix;
        int row, col;
        Matrix(){
                row = 0;
                col = 0;
        }
        Matrix(int _row, int _col) : row(_row), col(_col) {
                matrix.assign(row, vector<T>(col, 0));
```

```
    }

    Matrix<T> operator*(const Matrix<T> &other){
        Matrix<T> result(row, other.col);
        for(int i = 0; i < row; i++){
            for(int j = 0; j < other.col; j++){
                for(int k = 0; k < col; k++){
                    result.matrix[i][j] = (matrix[i][k] * other.matrix[k][j] + result.matrix[i][j]) % MOD;
                }
            }
        }
        return result;
    }

    Matrix<T> operator^(long long k){
        Matrix<T> result(row, row);
        for(int i = 0; i < row; i++){
            result.matrix[i][i] = 1;
        }
        Matrix<T> temp = (*this);
        for(; k > 0; k >>= 1, temp = temp * temp){
            if(k & 1) result = temp * result;
        }
        return result;
    }
};



//others function


void dfs(const vector<vector<int>> &G, vector<int> &d, vector<vector<int>> &p, int u){
    for(int v : G[u]){
        if(v == p[u][0]) continue;
        d[v] = d[u] + 1;
        p[v][0] = u;
        for(int i = 1; i < base; i++){
            p[v][i] = p[p[v][i - 1]][i - 1];
        }
        dfs(G, d, p, v);
    }
}
```

```
int lca(int u , int v, const vector<int> &d, const vector<vector<int>> &p){
    if(d[u] != d[v]){
        if(d[u] < d[v]) swap(u, v);

        int k = d[u] - d[v];

        for(int i = 0; (1 << i) <= k; i++){
            if(k & (1 << i)) u = p[u][i];
        }
    }

    if(u == v) return u;

    int h = __lg(d[u]);
    for(int i = h; i >= 0; i--){
        if(p[u][i] != p[v][i]){
            u = p[u][i];
            v = p[v][i];
        }
    }
    return p[u][0];
}

int main(){
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.tie(NULL);
    open();
    int n;
    cin >> n;
    vector<vector<int>> G(n + 1);
    for(int i = 1; i < n; i++){
        int a, b;
        cin >> a >> b;
        G[a].push_back(b);
        G[b].push_back(a);
    }

    vector<vector<int>> parents(n + 1, vector<int>(base, 0));
    vector<int> depth(n + 1);
    dfs(G, depth, parents, 1);
    int q;
    cin >> q;
    while(q--){
        int r, u, v;
        cin >> r >> u >> v;
        int a = lca(u, v, depth, parents);
        int b = lca(u, r, depth, parents);
        int c = lca(r, v, depth, parents);

        cout << (a ^ b ^ c) << endl;
    }

    return 0;
}
```

Output:



Subtask Info

Status: ✓ Correct Answer Submission ID: 85295467

Score: 100 | Time: 0.85s | Memory: 39.4M

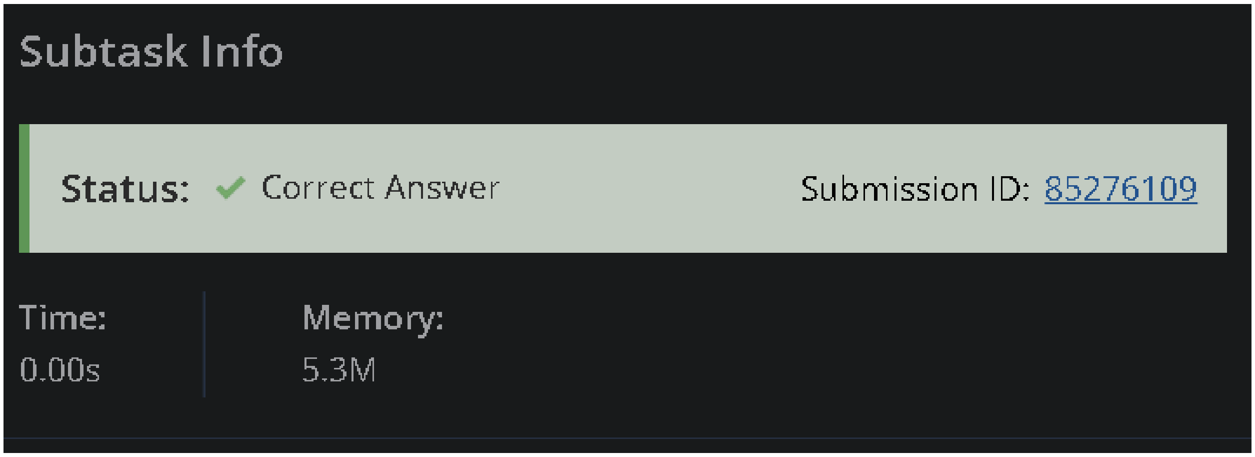
Sub-Task	Task #	Score	Result (time)
1	0	NA	AC (0.003895)
1	1	NA	AC (0.004084)
Final Score - 20.000000			**Result - AC**
2	2	NA	AC (0.006063)
2	3	NA	AC (0.008436)
2	4	NA	AC (0.028377)
2	5	NA	AC (0.275515)
Final Score - 40.000000			**Result - AC**
3	6	NA	AC (0.607809)

7. Cosmic Temple

    Code:

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

using namespace std;

void bfs(int temp,int n,vector<list<int>>adj){

    vector <bool> visited(n,false);

    int c = 0;

    vector <int> count(n,0);

    list<int> q;

    visited[temp] = true;

    count[temp]=0;

    q.push_back(temp);

    while(q.empty()==false){

        int sr = q.front();

        c = count[sr]+1;

        q.pop_front();

        for(auto adjc : adj[sr]){

            if(!visited[adjc]){

                visited[adjc] = true;

                count[adjc] = c;

                q.push_back(adjc);
            }
        }
    }
    int s = 0;
    for(int i=0;i<n;i++) s = s+count[i];
    cout<<s<<" ";
}
int main() {
          vector<list<int>>adj;
          int n;
          cin>>n;
          adj.resize(n);
          for(int i=0;i<n-1;i++){
              int v,u;
              cin>>v>>u;
              adj[v].push_back(u);
              adj[u].push_back(v);
          }
          for(int i=0;i<n;i++) bfs(i,n,adj);
          return 0;
}
```

Output:



Subtask Info

Status: ✓ Correct Answer

Submission ID: 85276109

Time:
0.00s

Memory:
5.3M

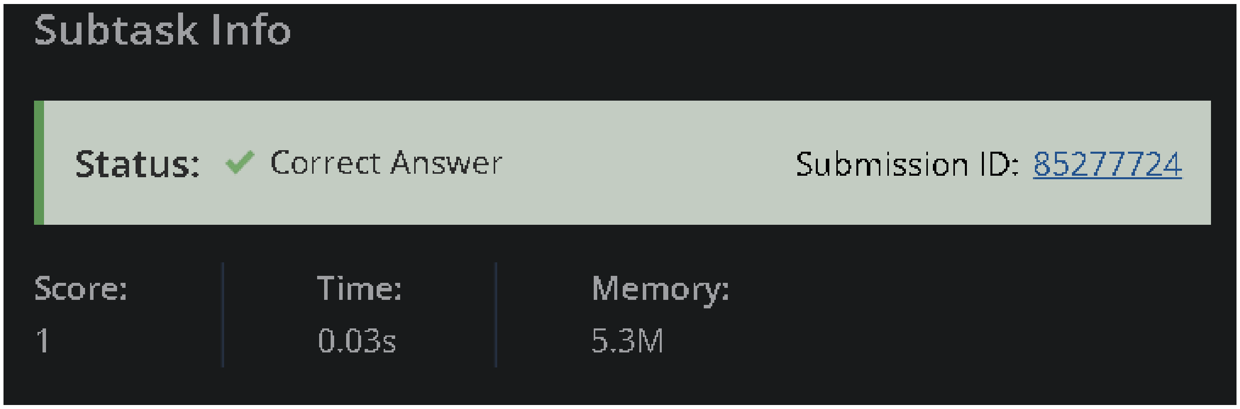
8. Secret Tree

Code:

```
#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]){

                                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();
}
```

Output:



Subtask Info

Status: ✓ Correct Answer

Submission ID: 85277724

Score:
1

Time:
0.03s

Memory:
5.3M