

Worksheet 6

Student Name:

Ravindra Singh Mehta

UID:20BCS7060

Branch: CSE

Section/Group:DWWC-43

Semester:5

Date of

Performance:10-01-
2023

Subject Name:Data
Structure

Subject Code:

1. Aim/Overview of the practical:

- i) Consider an infinite full binary tree (each node has two children except the leaf nodes) defined as follows. For a node labelled v its left child will be labelled $2*v$ and its right child will be labelled $2*v+1$. The root is labelled as 1.

You are given N queries of the form $i\ j$. For each query, you have to print the length of the shortest path between node labelled i and node labelled j .

- ii) You are given a rooted tree with NN nodes (numbered 11 through NN); node 11 is the root. Each node has a value; let's denote the value of node ii by A_i .

You may perform the following operation any number of times (including zero): choose any node which still exists in the tree and remove the whole subtree of this node including itself.

Let's define the *profit* as the sum of values of all nodes which currently exist in the tree minus $\sum kX \cdot k$, where k denotes the number of times this operation was performed. Find the maximum possible profit.

- iii) You are given an undirected tree having NN nodes. The color of each node is either Black or White.

In one step, you can choose a node uu , toggle the color of the chosen node uu , as well as toggle the colors of all the nodes vv , such that $\{u, v\}$ is an edge in the tree.

Can you find out the minimum number of steps required to make the color of all the nodes black?

Note: Since the input is large, prefer using fast input methods.

iv) You are given a family tree with NN members numbered $1 \dots N$. Every person ii , except for the founder of the family (root) has a parent denoted by $P[i]$. $P[\text{root}] = -1$ by definition. Person ii is a descendant of person jj if ii belongs to the subtree rooted at jj .

The net worth (adjusted for inflation) is given for all the members of the family. Your task is to find the maximum difference in net worth's of two members where one member is a descendant of the other.

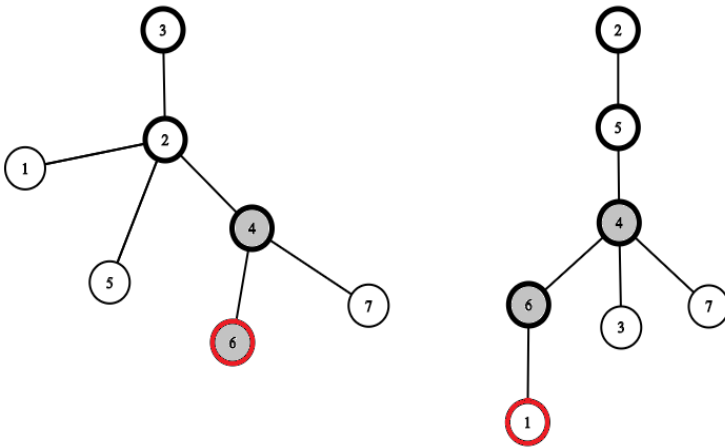
You can assume that no two members of this family married each other. So it remains a "family tree".

v) Ayush has two rooted trees. Each of them contains NN vertices (numbered 1 through NN); let's denote a vertex vv in tree 1 by v_1 and a corresponding vertex vv in tree 2 by v_2 .

For each tree tt , any vertex on the path from the root to a vertex v_{tv} (including v_{tv}) is an ancestor of v_{tv} and the depth of v_{tv} is the distance of v_{tv} from the root of tree tt .

For any integer ww ($1 \leq w \leq N$) and vertices u_1 and v_2 , ww is said to be a *common ancestor* of u_1 and v_2 if:

- in tree 1 , w_1 is an ancestor of u_1
- in tree 2 , w_2 is an ancestor of v_2
- the depths of vertices w_1 and w_2 (from their respective roots) are the same



For example, in the figure above, the pair $(u_1, v_2) = (6, 1)$ has two common ancestors — 4 and 6. Although vertex 2 is an ancestor for each of them, the depths of 2 and 4 are not the same.

Help Ayush find the maximum number of common ancestors for a pair of vertices, i.e. the largest integer cc such that there is a pair of vertices (u_1, v_2) (possibly $u = v$) with cc common ancestors.

vi) In a rooted tree, the (or LCA for short) of two vertices u and v is defined as the lowest vertex that is ancestor of both that two vertices.

Given a tree of N vertices, you need to answer the question of the form " $r \ u \ v$ " which means if the root of the tree is at r then what is LCA of u and v .

vii) Nearly $10^9 + 7$ trillion kilometers far away from the Milkyway, there exists a parallel Universe of Light called the Cosmos of Solon where each planet is called a Temple. There are total NN temples. Unlike humans, the Solomons (Giant aliens of the Light World) can travel from one temple to another through Cosmic Bridges and the distance observed by them between any two temples is 00 units (Interesting? Well, they travel with more than the speed of light)

But the year 2020, proved to be a curse for Solomons. This year, a Mega Star Urus struck their Universe and damaged many Cosmic Bridges. However the God of Light ensured that for any two distinct temples uu and vv there exists exactly one unique path through Cosmic Bridge(s), but the distance between two directly connected temples through a Bridge has now increased to 11 units.

To prevail the existence of this Universe, the God of Light carved himself into exactly NN Solomons which now rule the NN temples and are independent of each other. One by one, the i^{th} Solomon ($0 \leq i \leq N-1$) transmits Light pulses through the network of Cosmic Bridges and determines the sum of distances to the remaining $N-1$ Solomons.

Can you help them find these quantities so they can reunite into God of Light again?

xiii) Abdullah has recently moved to Tolland. In Tolland, there are NN cities (numbered 11 through NN) connected by $N-1$ bidirectional roads such that it is possible to visit any city from any other city. For each city ii , there is a *toll value* TL_i . Abdullah lives in city XX . He has planned a tour of Tolland lasting for $2N$ days. For each ii ($1 \leq i \leq N$), on the $2i-1$ -th day, he travels from city XX to city ii , and on the $2i$ -th day, he travels from city ii back to city XX .

In Tolland, there are also special discount coupons. For each city CC and positive integer VV , there is a discount coupon for city CC with *value* VV . All discount coupons may be bought anywhere in Tolland; the cost of each discount coupon is twice its value. When travelling from any city ii to any (not necessarily different) city jj , Abdullah must pay a toll exactly once in each city on the path between them (including ii and jj); even if he passed through some cities before, he must pay the toll in these cities as well. For any city ii , the toll is TL_i when not using any discount coupons; however, if Abdullah bought some discount coupons for city ii , he may use some or all of them to decrease this toll to $\max(0, TL_i - V)$, where V is the sum of values of coupons for city ii used by Abdullah. Each coupon may be used at most once per day, but they do not expire when used and the same coupon may be used on multiple days. On each day, Abdullah may use any coupons he wants, but a coupon for city ii may only be used to decrease the toll in city ii .

Abdullah has a travel budget: on each day, he must not spend more than KK units of money on paying tolls. Before the start of the tour, he may choose to buy any number of discount coupons, for any cities and with any values (possibly different ones for different cities); the cost of coupons is not included in the budget for any day of the tour. Help Abdullah find the minimum total amount of money he must spend on discount coupons so that on each day during the tour, he will be able to stay within his budget by using the coupons he bought.

xiv) A tree is an undirected connected graph without any cycles. You are given a tree with N nodes and a parameter K . Each node in the tree has a positive weight $W[i]$. The tree needs to be partitioned into at most K subtrees by removing at most $K-1$ edges such that the maximum weight of the resulting subtrees is as small as possible. Find that smallest value of the maximum subtree weight that can be obtained by removing at most $K-1$ edges.

xv) The apocalyptic demon [Kali](#) is on a rampage. The ground shudders under his feet, trees shrivel and animals and birds scurry away from his path. In order to save the universe from devastation, all the devtas led by devraj [Indra](#) decide to meditate to please Lord [Vishnu](#) so that he appears in the form of [Kalki](#), his last avatar, and kill the demon. Each devta can sit in meditation at a particular place on [Swarglok](#) (the heaven). Considering Swarglok as a 2-dimensional plane, the position of each devta is a fixed point with integer coordinates. The positions of all the devtas are distinct from each other.

While meditating the devtas connect to each other by means of astral projections - metaphysical threads that connect one devta to another. They must do so in such a way to satisfy two criteria:

- Each devta must be able to connect to every other devta by following a path of astral projections.
- No subgroup of devtas may be connected by a cycle of astral projections.

In simple terms, the astral projections must form a tree connecting all devtas.

What needs to be optimized ?

Once the devtas have taken their positions and are connected as a single group to start meditation, a *ring of influence* is formed around each devta. This ring is centered at the position of the devta, and has a radius equal to the Euclidean distance from this devta to the furthest devta directly connected via an astral projection.

Since different devtas have their own supernatural powers ([Indra](#) is the rain god, [Agni](#) is the fire god, [Varuna](#) is the water god, [Vayu](#) is the air god , etc), the influence ring of each devta has an adverse effect on all other devtas which lie within or on this ring. In other words, the efficiency of a devta to perform meditation decreases as the number of influence rings that include him increases. For each devta D_i , define C_i as the number of influence rings that include or touch D_i (including their own). Now devraj Indra wants the devtas to connect in such a manner so as to minimize the maximum value of C_i over all devtas. So he has sent a message to [Bhulok](#) (the earth) and he needs help from the best programmers on the planet.

2. Steps for experiment/practical/Code:

i)

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

```
int main() {
    int t;
    cin>>t;
    while(t-->0) {
        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;
}
```

ii)

```
#include "bits/stdc++.h"

using namespace std;
typedef long long ll;
typedef long double ld;
typedef pair<ll,ll> pll;
typedef unsigned long long ull;
#define all(x) (x).begin(), (x).end()

const int MOD = 1000000007; // 1000000009 , 1000000023 , 1000000007 , 998244353
const ll INF = ll(1e18);
const ld PI = 4*atan((ld)1);
const char nl = '\n';

int main() {
    ios_base::sync_with_stdio(false);
    cin.tie(NULL);
    cout.precision(10);

    ll tt;
    cin >> tt;
    while(tt--){
        ll n,x; cin >> n >> x;
        vector<vector<ll>>g(n + 1 , vector<ll>());
        vector<ll>val(n);
        for(auto &x:val)cin >> x;
        for(ll i=0 ; i<n-1 ; i++){
            ll u,v; cin >> u >> v;
            g[u].push_back(v);
            g[v].push_back(u);
        }
        vector<ll>dp(n+1);
```

```
auto dfs = [&](ll s, ll p, auto && dfs)->ll{
    ll res = val[s-1];
    for(auto x:g[s]){
        if(x != p)res += dfs(x, s, dfs);
    }
    return dp[s] = max(res, -x);
};
cout << dfs(1, -1, dfs) << nl;
}
}
```

iii)

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

```

iv)

```
#include <iostream>
#include <bits/stdc++.h>
#include <cmath>
#include <algorithm>
#include <vector>
#include <deque>
#include <set>
#include <stack>
#include <string>
#include <map>
#include <queue>

//#pragma comment(linker, "/stack:2000000000")
//#pragma GCC optimize("Ofast")
//#pragma GCC target("sse,sse2,sse3,ssse3,sse4,popcnt,abm,mmx,avx,avx2,tune=native")

using namespace std;

#define int long long
#define ll long long
#define vi vector<long long>
#define pb push_back
#define sz(s) (int)s.size()
#define all(v) v.begin(), v.end()
#define show(a) cerr << #a << " -> " << a << "\n"
#define pp pair<int,int>
#define FF first
#define SS second
#define endl "\n"
#define ld long double
//typedef long double ld;

const int N = 1e6 + 2, N3 = 1e3 + 6, inf = 1e9 + 7, LOG = 20;
map<char, int> md{ {'N', 0}, {'E', 1}, {'S', 2}, {'W', 3} };
const int dx[] = {-1, 0, 1, 0};
const int dy[] = {0, 1, 0, -1};
string stepDir = "RDLU";

int n, w[N], dp[N], root, mn[N], mx[N], ans;
vi g[N];
```

```

void dfs(int v, int p) {
    mn[v] = w[v], mx[v] = w[v];
    for (auto to : g[v]) {
        if (to == p) continue;
        dfs(to, p);
        mn[v] = min(mn[v], mn[to]);
        mx[v] = max(mx[v], mx[to]);
    }
}

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

    //    freopen("triangles.in", "r", stdin);
    //    freopen("triangles.out", "w", stdout);

    cin >> n;
    for (int i = 1; i <= n; i++) {
        cin >> w[i];
    }
    for (int i = 1; i <= n; i++) {
        int p;
        cin >> p;
        if (p == -1) {
            root = i;
            continue;
        }
        g[p].pb(i);
    }

    dfs(root, 0);
    for (int i = 1; i <= n; i++) {
        ans = max(ans, max(abs(w[i] - mn[i]), abs(w[i] - mx[i]))));
    }
    cout << ans;

    return 0;
}

```

v)

```
#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;
}

```

vi)

```
#include<bits/stdc++.h>
using namespace std;
int sp[200100][18];
int depth[200100];int n;
vector <int> g[200100];
void dfs(int u,int pre){
    sp[u][0] = pre;
    for (int j = 1;j < 18;j ++){
        sp[u][j] = sp[sp[u][j-1]][j-1];
    }
    depth[u] = depth[pre] + 1;
    for (auto v:g[u]){
        if (v == pre)continue;
        dfs(v,u);
    }
}
int lca(int x,int y){
    if (depth[x] < depth[y])swap(x,y);
    for (int j = 17;j >= 0;j --){
        if (depth[x] - (1<<j) >= depth[y]){
            x = sp[x][j];
        }
    }
    for (int j = 17;j >= 0;j --){
        if (sp[x][j] != sp[y][j]){
            x = sp[x][j];
            y = sp[y][j];
        }
    }
    if (x != y){
        x = sp[x][0];
    }
    return x;
}
int best(int x,int y){
    if (depth[x] > depth[y])return x;
    return y;
}
int main(){
    ios_base::sync_with_stdio(0);cin.tie(nullptr);cout.tie(nullptr);
```

```

cin>>n;
for (int i = 1;i < n;i ++){
    int u,v;
    cin>>u>>v;
    g[u].push_back(v);
    g[v].push_back(u);
}
dfs(1,1);
int q;
cin>>q;
while (q--){
    int r,u,v;
    cin>>r>>u>>v;
    cout<<best(best(lca(r,u),lca(r,v)),lca(u,v))<<'\n';
}
return 0;
}

```

vii)

```

#include <iostream>
#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() {
    // your code goes here

    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;
}

```

```

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

inline int read() {
int x = 0, f = 1; char ch = getchar();
while (ch < '0' || ch > '9') { if (ch == '-') f = -1; ch = getchar(); }
while (ch >= '0' && ch <= '9') { x = x * 10 + ch - 48; ch = getchar(); }
return x * f;
}

inline ll readl() {
ll x = 0, f = 1; char ch = getchar();
while (ch < '0' || ch > '9') { if (ch == '-') f = -1; ch = getchar(); }
while (ch >= '0' && ch <= '9') { x = x * 10 + ch - 48; ch = getchar(); }
return x * f;
}

const int N = 1e4 + 10;
vector<int> G[N];
bool vis[N];
ll dis[N], a[N], ans, k, n, p;

ll dfs1(int u, int fa) {
ll res = 0;
for (auto v : G[u]) {
if (v != fa) res = max(res, dfs1(v, u));
}
return dis[u] = res + a[u];
}

void dfs2(int u, int fa) {
if (dis[u] <= k) return;
if (dis[u] - a[u] <= k) {
ans += dis[u] - k;
return;
}

ans += a[u];
for (auto v : G[u]) {
if (v != fa) dfs2(v, u);
}}

```

```
int main() {
int T = read();
while (T--) {
n = read(), p = read(), k = readl();

for (int i = 1; i <= n; i++) {
G[i].clear();
a[i] = readl();
}

for (int i = 1; i < n; i++) {
int u = read(), v = read();
G[u].emplace_back(v);
G[v].emplace_back(u);
}

dfs1(p, -1);
ans = 0;
dfs2(p, -1);

printf("%lld\n", ans * 2);
}
return 0;
}
```

xiv)

```
#include <bits/stdc++.h>
using namespace std;
using ll = long long;

ll n, k, sum = 0;
ll cntk = 0, mxwt = -1;
vector<ll> wt_node;
vector<ll> wt_stree;
vector<vector<ll>> g;

void dfs(ll nn, ll par, ll x)
{
    vector<ll> wt_child;
    wt_stree[nn] = wt_node[nn];
    for(auto it : g[nn])
        if(it != par)
            dfs(it, nn, x), wt_child.push_back(wt_stree[it]);

    ll sz = wt_child.size();
    sort(wt_child.begin(), wt_child.end());
    for(ll i=0; i<sz; i++)
    {
        if(wt_stree[nn]+wt_child[i] <= x)
            wt_stree[nn] += wt_child[i];
        else
        {
            cntk += sz - i;
            break;
        }
    }

    bool check(ll x)
    {
        cntk = 0;
        dfs(1, 0, x);
        if(cntk <= k-1)
            return 1;
        return 0;
    }

    void solve()
    {
```

```

cin >> n >> k;
wt_node.assign(n+1, 0);
g.assign(n+1, {});
wt_stree.assign(n+1, 0);

ll i, x, y;
for(i=1; i<=n; i++)
{
cin >> wt_node[i];
sum += wt_node[i];
mxwt = max(mxwt, wt_node[i]);
}
for(i=1; i<n; i++)
{
cin >> x >> y;
g[x].push_back(y);
g[y].push_back(x);
}

ll lo = mxwt, hi = sum;
ll mid, ans;
while(lo <= hi)
{
mid = (lo+hi)/2;
if(check(mid))
{
ans = mid;
hi = mid-1;
}
else
lo = mid+1;
}

cout << ans << "\n";
}

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

ll t = 1;
// cin >> t;
while(t--) solve();
return 0;}

```

xv)

```
#include<bits/stdc++.h>
using namespace std;
typedef long long LL;

void FastIO(void)
{
ios_base::sync_with_stdio(false);
cin.tie(NULL);
}

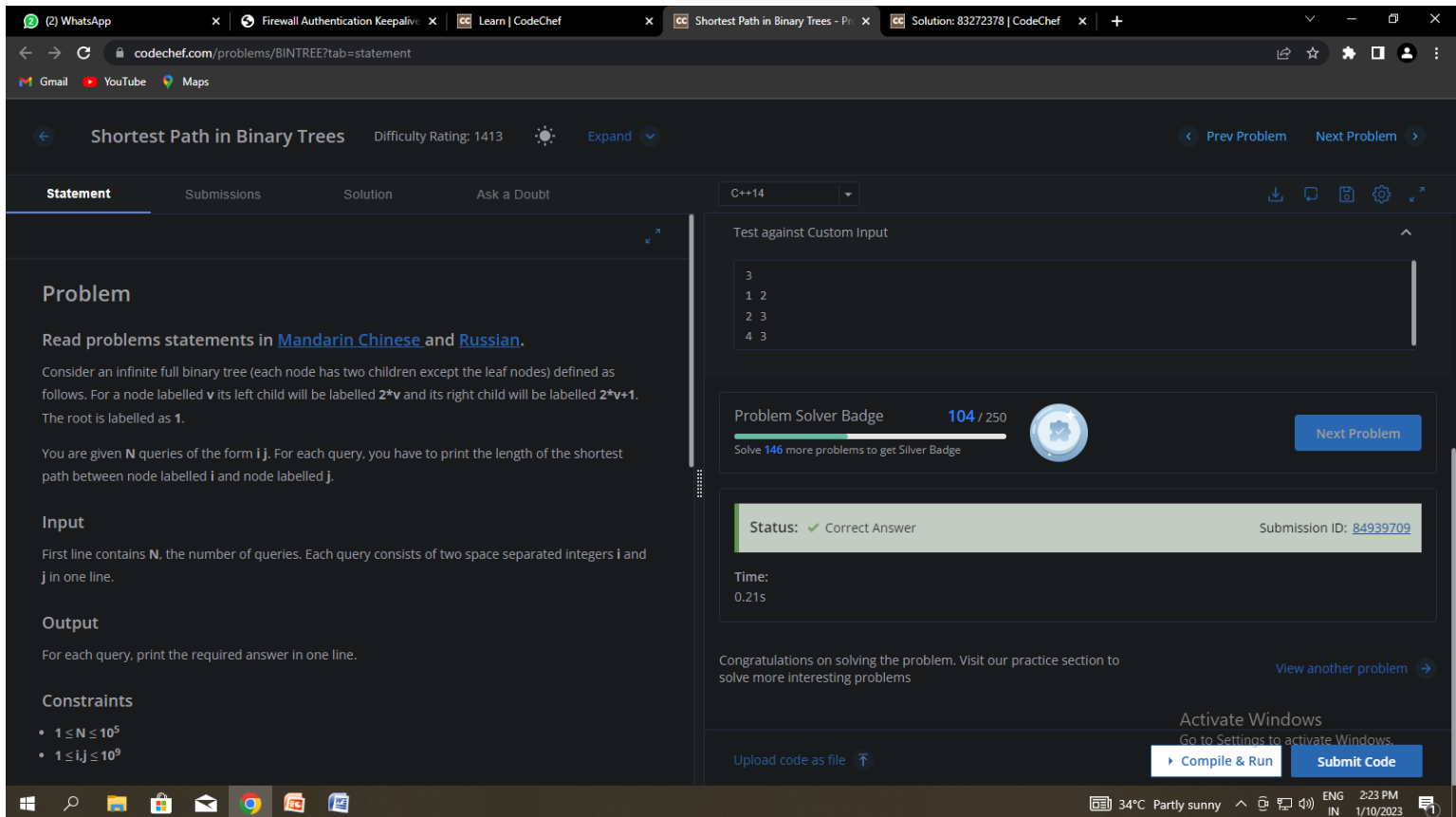
signed main()
{
FastIO();
LL T;
cin >> T;
while(T--)
{
LL N;
cin >> N;

vector<vector<LL>> coOrd;
for(LL i = 1; i <= N; ++i)
{
LL x, y;
cin >> x >> y;
coOrd.push_back({x, y, i});
}
sort(coOrd.begin(), coOrd.end());

for(LL i = 0; i < N-1; ++i)
cout << coOrd[i][2] << " " << coOrd[i+1][2] << endl;
}
}
```

3. Observations/Discussions/ Complexity Analysis:

i)



The screenshot shows a web browser window with the CodeChef website. The page displays the problem "Shortest Path in Binary Trees" with a difficulty rating of 1413. The problem statement describes an infinite full binary tree where the root is labeled 1, and for a node labeled v , its left child is labeled $2*v$ and its right child is labeled $2*v+1$. The task is to find the shortest path between two nodes i and j for N queries. The input consists of N queries, each with two integers i and j . The output is the length of the shortest path for each query. The constraints are $1 \leq N \leq 10^5$ and $1 \leq i, j \leq 10^9$.

The solution is implemented in C++14. The code uses a function `findLCA` to find the Lowest Common Ancestor (LCA) of two nodes i and j . The LCA is found by repeatedly dividing the larger of the two nodes by 2 until they are equal. The length of the shortest path is then calculated as the sum of the number of steps from i to the LCA and from the LCA to j .

The test against custom input shows the following input:

```
3
1 2
2 3
4 3
```

The problem solver badge shows a score of 104 / 250. The status is "Correct Answer" with a submission ID of 84939709. The time taken is 0.21s.

The bottom of the screenshot shows the Windows taskbar with the date and time as 1/10/2023, 2:23 PM, and the temperature as 34°C.

ii)

Subtree Removal

Difficulty Rating: 2010

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

Let's define the *profit* as the sum of values of all nodes which currently exist in the tree minus $X \cdot k$, where k denotes the number of times this operation was performed. Find the maximum possible profit.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains two space-separated integers N and X .
- The second line contains N space-separated integers A_1, A_2, \dots, A_N .
- Each of the following $N - 1$ lines contains two space-separated integers u and v denoting that nodes u and v are connected by an edge.

Output

For each test case, print a single line containing one integer — the maximum profit.

Constraints

- $1 \leq T \leq 10$
- $1 \leq N \leq 10^5$
- $1 \leq X \leq 10^9$
- $1 \leq u, v \leq N$
- $|A_i| \leq 10^9$ for each valid i
- the graph described on the input is a tree

C++14

```
1
3 5
1 -5 -10
1 2
```

Problem Solver Badge

105 / 250

Next Problem

Status: Correct Answer

Submission ID: [84940636](#)

Time: 0.46s

Sub-Task	Task #	Result (time)
1	1	AC (0.003799)
1	2	AC (0.003871)
1	3	AC

Upload code as file

Compile & Run

Submit Code

Windows Taskbar

System Tray

iii)

(2) WhatsApp

Firewall Authentication Keepalive

Black and White Tree - Problem

Black and White Tree Submission

Solution: 84923596 | CodeChef

+

codechef.com/problems/BAWTREE?tab=statement

🔍

☆

⚙️

👤

⋮

Gmail

YouTube

Maps

⏪

Black and White Tree

Difficulty Rating: 3008

☀️

Expand

⏩

Prev Problem

Next Problem

Statement

Submissions

Solution

Problem

You are given an undirected tree having N nodes. The color of each node is either Black or White.

In one step, you can choose a node u , toggle the color of the chosen node u , as well as toggle the colors of all the nodes v , such that $\{u, v\}$ is an edge in the tree.

Can you find out the minimum number of steps required to make the color of all the nodes black?

Note: Since the input is large, prefer using fast input methods.

Input Format

- The first line of input contains a single integer T , denoting the number of test cases. The description of the T test cases follows.
- The first line of each test case contains a single integer N .
- The second line of each test case contains N space separated integers A_1, A_2, \dots, A_N . $A_i = 0$ represents that the initial color of the i^{th} node is black. $A_i = 1$ represents that the initial color of the i^{th} node is white.
- The following $N - 1$ lines contain two space-separated integers x and y , which denotes that there is an edge between nodes x and y .

Output Format

For each test case, output in a single line the minimum number of steps required to make the color of

C++14

Problem Solver Badge

100 / 250

Solve 144 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer

Submission ID: 84941092

Time: 0.15s

Sub-Task	Task #	Result (time)
1	1	AC (0.029013)
1	2	AC (0.009280)
1	3	AC (0.084825)
1	4	AC (0.085115)
1	5	AC (0.140123)
1	6	AC

Upload code as file

Compile & Run

Submit Code

Windows Taskbar

34°C Partly sunny

ENG IN

2:38 PM

1/10/2023

egov

✉ egov@cumail.in

iv)

(2) WhatsApp

Firewall Authentication Keepalive

Family Tree - Problems | CodeChef

Family Tree Submissions | CodeChef

Solution: 84941494 | CodeChef

+

codechef.com/problems/FAMTREE?tab=statement

GmailYouTubeMaps

Family Tree

Difficulty Rating: NA

Expand

Statement

Submissions

Solution

Problem

You are given a family tree with N members numbered $1 \dots N$. Every person i , except for the founder of the family (root) has a parent denoted by $P[i]$. $P[\text{root}] = -1$ by definition. Person i is a descendant of person j if i belongs to the subtree rooted at j .

The net worth (adjusted for inflation) is given for all the members of the family. Your task is to find the maximum difference in net worth's of two members where one member is a descendant of the other.

You can assume that no two members of this family married each other. So it remains a "family tree".

###Input:

- First line will contain an integer N , denoting the number of members.
- Second line contains N space separated integers $W[1], W[2] \dots W[N]$ denoting the net_worth of the N members.
- Third line contains N space separated integers $P[1], P[2] \dots P[N]$ denoting the parents of the N members.

###Output: Output in a single line the maximum difference in net worth's of any two members where one member is a descendant of the other.

###Constraints

- $2 \leq N \leq 10^5$
- $-10^8 \leq W[i] \leq 10^8$

C++14

Test against Custom Input

Your program will run with no input

Problem Solver Badge

107 / 250

Solve 143 more problems to get Silver Badge

Status: ✓ Correct Answer

Submission ID: [84941645](#)

Time: 0.04s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows

Upload code as file

Compile & Run

Submit Code

Windows Taskbar

34°C Partly sunny

ENG IN

2:44 PM

1/10/2023

Common Ancestors

Difficulty Rating: 2539

Expand

Statement

Submissions

Solution

Problem

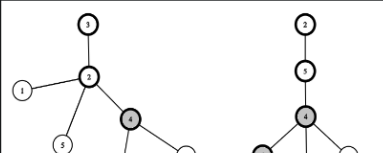
Read problems statements in [Hindi](#), [Mandarin Chinese](#), [Russian](#), [Vietnamese](#), and [Bengali](#) as well.

Ayush has two rooted trees. Each of them contains N vertices (numbered 1 through N); let's denote a vertex v in tree 1 by v_1 and a corresponding vertex v in tree 2 by v_2 .

For each tree t , any vertex on the path from the root to a vertex v_t (including v_t) is an ancestor of v_t and the depth of v_t is the distance of v_t from the root of tree t .

For any integer w ($1 \leq w \leq N$) and vertices u_1 and v_2 , w is said to be a *common ancestor* of u_1 and v_2 if:

- in tree 1, w_1 is an ancestor of u_1
- in tree 2, w_2 is an ancestor of v_2
- the depths of vertices w_1 and w_2 (from their respective roots) are the same



C++14

Test against Custom Input

```
3
1
-1
-1
```

Problem Solver Badge

108 / 250

Solve 142 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer

Submission ID: [84942027](#)

Time: 1.18s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows.

Upload code as file

Compile & Run

Submit Code

WhatsApp

Firewall Authentication Keepalive

Common Ancestors - Problems

Common Ancestors Submissions

Solution: 84922736 | CodeChef

codechef.com/problems/TREEANC?tab=statement

Gmail

YouTube

Maps

Common Ancestors

Difficulty Rating: 2539

Expand

Prev Problem

Next Problem

Statement

Submissions

Solution

Problem

Read problems statements in Hindi, Mandarin Chinese, Russian, Vietnamese, and Bengali as well.

Ayush has two rooted trees. Each of them contains N vertices (numbered 1 through N); let's denote a vertex v in tree 1 by v1 and a corresponding vertex v in tree 2 by v2.

For each tree t, any vertex on the path from the root to a vertex vt (including vt) is an ancestor of vt and the depth of vt is the distance of vt from the root of tree t.

For any integer w (1 ≤ w ≤ N) and vertices u1 and v2, w is said to be a common ancestor of u1 and v2 if:

in tree 1, w1 is an ancestor of u1

in tree 2, w2 is an ancestor of v2

the depths of vertices w1 and w2 (from their respective roots) are the same

Diagram showing two rooted trees. Tree 1 has root 1, children 2 and 3, and child 4 for node 2. Tree 2 has root 2, children 3 and 4, and child 5 for node 3. Nodes 1, 2, 3, 4 are in tree 1 and nodes 2, 3, 4, 5 are in tree 2.

C++14

Test against Custom Input

3

1

-1

-1

Problem Solver Badge

108 / 250

Solve 142 more problems to get Silver Badge

Next Problem

Status: ✓ Correct Answer

Submission ID: 84942027

Time: 1.18s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows.

Upload code as file

Compile & Run

Submit Code

34°C Partly sunny

ENG IN

2:48 PM

1/10/2023

✉ egov@cumail.in

codechef.com/problems/TALCA?tab=statement

Lowest Common Ancestor Difficulty Rating: 2600 Expand

Statement Submissions Solution

C++14

Problem

Read problems statements in [Mandarin Chinese](#).

In a rooted tree, the (or LCA for short) of two vertices u and v is defined as the lowest vertex that is ancestor of both that two vertices.

Given a tree of N vertices, you need to answer the question of the form "r u v" which means if the root of the tree is at r then what is LCA of u and v .

Input

The first line contains a single integer N . Each line in the next $N - 1$ lines contains a pair of integer u and v representing a edge between this two vertices.

The next line contains a single integer Q which is the number of the queries. Each line in the next Q lines contains three integers r, u, v representing a query.

Output

Problem Solver Badge 109 / 250

Solve 141 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: 84942680

Time: 0.26s

Sub-Task	Task #	Score	Result (time)
1	0	NA	AC (0.005287)
1	1	NA	AC (0.005653)
Final Score - 20.000000		Result - AC	

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 2:54 PM 1/10/2023

vii)

codechef.com/problems/COSTEMP?tab=statement

Cosmic Temple Difficulty Rating: NA Expand

Statement Submissions Solution

C++14

Problem

Nearly $10^9 + 7$ trillion kilometers far away from the Milkyway, there exists a parallel Universe of Light called the Cosmos of Solon where each planet is called a Temple. There are total N temples. Unlike humans, the Solomons (Giant aliens of the Light World) can travel from one temple to another through Cosmic Bridges and the distance observed by them between any two temples is 0 units (Interesting? Well, they travel with more than the speed of light)

But the year 2020, proved to be a curse for Solomons. This year, a Mega Star Urus struck their Universe and damaged many Cosmic Bridges. However the God of Light ensured that for any two distinct temples u and v there exists exactly one unique path through Cosmic Bridge(s), but the distance between two directly connected temples through a Bridge has now increased to 1 units.

To prevail the existence of this Universe, the God of Light carved himself into exactly N Solomons which now rule the N temples and are independent of each other. One by one, the i^{th} Solomon ($0 \leq i \leq N - 1$) transmits Light pulses through the network of Cosmic Bridges and determines the sum of distances to the remaining $N - 1$ Solomons.

Can you help them find these quantities so they can reunite into God of Light again?

###Input Format:

- The first line of the input contains N , the total no. of temples.
- Each of the next $N - 1$ lines contain two space-separated integers u and v ($u \neq v$) denoting the bidirectional Cosmic Bridge between temples u and v .

Test against Custom Input

```
0 2
0 3
1 3
3 4
```

Problem Solver Badge 110 / 250

Solve 140 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: 84943448

Time: 0.00s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Upload code as file

Compile & Run Submit Code

34°C Partly sunny 3:06 PM 1/10/2023

xiii)

Abdullah and Toll Discounts

Difficulty Rating: NA

Expand

Statement

Submissions

Solution

discount coupon for city c with value v . All discount coupons may be bought anywhere in Tolland; the cost of each discount coupon is twice its value.

When travelling from any city i to any (not necessarily different) city j , Abdullah must pay a toll exactly once in each city on the path between them (including i and j); even if he passed through some cities before, he must pay the toll in these cities as well. For any city i , the toll is TL_i when not using any discount coupons; however, if Abdullah bought some discount coupons for city i , he may use some or all of them to decrease this toll to $\max(0, TL_i - V)$, where V is the sum of values of coupons for city i used by Abdullah. Each coupon may be used at most once per day, but they do not expire when used and the same coupon may be used on multiple days. On each day, Abdullah may use any coupons he wants, but a coupon for city i may only be used to decrease the toll in city i .

Abdullah has a travel budget: on each day, he must not spend more than K units of money on paying tolls. Before the start of the tour, he may choose to buy any number of discount coupons, for any cities and with any values (possibly different ones for different cities); the cost of coupons is not included in the budget for any day of the tour. Help Abdullah find the minimum total amount of money he must spend on discount coupons so that on each day during the tour, he will be able to stay within his budget by using the coupons he bought.

Input

- The first line of the input contains a single integer T denoting the number of test cases. The description of T test cases follows.
- The first line of each test case contains three space-separated integers N , X and K .
- The second line contains N space-separated integers TL_1, TL_2, \dots, TL_N .

Each of the next $N - 1$ lines contains two space-separated integers U and V denoting that cities U and V are connected by a bidirectional road.

C++14

```
2
4 2 6
4 7 6 2
2 1
```

Status : Successfully executed

Time: 0.006975 secs

Memory: 5.364 Mb

Input

```
4
4 2 6
4 7 6 2
2 1
2 3
2 4
3 1 8
2 3 1
1 2
2 3
```

Upload code as file

Compile & Run

Submit Code

Activate Windows

Go to Settings to activate Windows.

34°C Partly sunny

ENG IN

3:11 PM

1/10/2023

xiv)

codechef.com/problems/TREE02

Break the tree Difficulty Rating: NA Expand

Statement Submissions Solution

Problem

A tree is an undirected connected graph without any cycles. You are given a tree with N nodes and a parameter K . Each node in the tree has a positive weight $W[i]$. The tree needs to be partitioned into at most K subtrees by removing at most $K-1$ edges such that the maximum weight of the resulting subtrees is as small as possible. Find that smallest value of the maximum subtree weight that can be obtained by removing at most $K-1$ edges.

Input

First Line contains N , the number of nodes in the tree and K , the number of subtrees. Next line contains N integers, denoting weights of the nodes. The i th integer denotes $W[i]$. Next, there are $N-1$ lines with 2 integers each, denoting the edges in the tree.

Output

Output a single integer, the answer.

Constraints

- $1 \leq N \leq 20000$
- $1 \leq K \leq N$
- $1 \leq W[i] \leq 10^9$

Test against Custom Input

```
3 2
5 10 2
1 2
2 3
```

Problem Solver Badge 111 / 250

Solve 139 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: [84944932](#)

Time: 0.07s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows

Upload code as file

Compile & Run Submit Code

34°C Partly sunny ENG 3:19 PM IN 1/10/2023

xv)

codechef.com/problems/KALKI

Kali and Devtas Difficulty Rating: 3660 Expand

Statement Submissions Solution

Scoring and test data generation

The score is C , where $C = \max(C_i)$ for $1 \leq i \leq N$. The sequence of positions for devtas is generated randomly and the positions are distributed uniformly.

Example

Input:

```
1
5
-3 0
-1 0
0 1
1 0
2 0
```

Example of a valid output:

```
1 2
2 3
3 4
4 5
```

Explanation

The second devta is connected to two other devtas: the first devta (a distance of 2 away), and the third devta (a distance of $\sqrt{2}$ away). The larger of these distances is 2, so the radius of the second devta's ring of influence is 2. We can calculate the other rings similarly.

Here is a diagram with the astral projections represented in red, and all of the influence rings

Test against Custom Input

```
-1 0
0 1
1 0
2 0
```

Problem Solver Badge 112 / 250

Solve 138 more problems to get Silver Badge

Status: ✓ Correct Answer Submission ID: [84945530](#)

Time: 0.06s

Congratulations on solving the problem. Visit our practice section to solve more interesting problems

View another problem

Activate Windows

Go to Settings to activate Windows

Upload code as file

Compile & Run Submit Code

34°C Partly sunny ENG 3:24 PM IN 1/10/2023

Evaluation Grid (To be created as per the SOP and Assessment guidelines by the faculty):

Sr. No.	Parameters	Marks Obtained	Maximum Marks
1.			
2.			
3.			