

**UNIVERSITY INSTITUTE OF ENGINEERING**

**Department of Computer Science & Engineering**

**Subject Name:** Competitive coding-I

**Subject Code:** 20CSP-314

|  |  |
| --- | --- |
| **Submitted By:** Kanishk Soni **Submitted To: Mr. Syed Abdul Basit** | |
| **Subject Name** | CC LAB |
| **Subject Code** | 20CSP-314 |
| **Branch** | Computer Science |
| **Semester** | 5th |

**Practical Evaluation Sheet**

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **Sr.No** | **Program** | **Date** | **Evaluation** | | | | **Sign** |
| **Conduct**  **(12)** | **Viva(8)** | **Worksheet(10)** | **Total (30)** |
| 1 | Concept of Arrays |  |  |  |  |  |  |
| 2 | Concept of Stack and Queues |  |  |  |  |  |  |
| 3 | Concept of Linked List |  |  |  |  |  |  |
| 4 | Concept of Searching and Sorting |  |  |  |  |  |  |
| 5 | Concept of Graph |  |  |  |  |  |  |
| 6 | Concept of Trees |  |  |  |  |  |  |
| 7 |  |  |  |  |  |  |  |
| 8 |  |  |  |  |  |  |  |
| 9 |  |  |  |  |  |  |  |
| 10 |  |  |  |  |  |  |  |

**Experiment 6**

**Student Name: Kanishk Soni UID: 20BCS9398**

**Branch: CSE Section/Group:707\_WM\_B**

**Semester: 5th Date of Performance: 17/10/2022**

**Subject Name: Competitive Coding - I Subject Code: 20CSP-314**

**Question1: Tree-huffman-decoding**

**Code:**

void decode\_huff(node\* root, string s) {

node \*troot = root;

int i=0,k=0;

while (s[i] != '\0') {

if (s[i] == '1' ) {

root = root->right;

if (root->data != '\0') {

cout<<root->data;

root = troot;

}

i++;

} else {

root=root->left;

if (root->data != '\0'){

cout<<root->data;

root = troot;

}

i++;

}

}

}



**Question 2:** **Balanced-forest problem**

**Code:**

#include <bits/stdc++.h>

using namespace std;

typedef long long ll;

const ll INF = (ll) 1e18;

const int N = (int) 5e4 + 10;

vector<int> g[N];

ll c[N];

ll f[N];

ll res = INF;

ll tot = 0;

bool was[N];

void upd(ll a, ll b, ll c) {

if (a == b && c <= a)

res = min(res, a - c);

if (a == c && b <= a)

res = min(res, a - b);

if (b == c && a <= b)

res = min(res, b - a);

}

set<ll>\* unite(set<ll>\* a, set<ll>\* b) {

if (a->size() > b->size())

swap(a, b);

for (ll x : \*a) {

if (b->count(tot - 2 \* x))

upd(tot - 2 \* x, x, x);

if (b->count(x))

upd(x, x, tot - 2 \* x);

if ((tot - x) % 2 == 0 && b->count((tot - x) / 2))

upd((tot - x) / 2, x, (tot - x) / 2);

}

for (ll x : \*a) {

b->insert(x);

}

delete a;

return b;

}

set<ll>\* dfs(int v) {

was[v] = true;

f[v] = c[v];

set<ll>\* sv = new set<ll>();

for (int to : g[v])

if (!was[to]) {

set<ll>\* sto = dfs(to);

f[v] += f[to];

sv = unite(sv, sto);

}

if (f[v] % 2 == 0 && sv->count(f[v] / 2))

upd(f[v] / 2, f[v] / 2, tot - f[v]);

if (sv->count(tot - f[v]))

upd(tot - f[v], 2 \* f[v] - tot, tot - f[v]);

if (sv->count(2 \* f[v] - tot))

upd(2 \* f[v] - tot, tot - f[v], tot - f[v]);

sv->insert(f[v]);

return sv;

}

void solve() {

int n;

cin >> n;

for (int i = 0; i < N; i++) {

was[i] = false;

g[i].clear();

c[i] = 0;

}

tot = 0;

res = INF;

for (int i = 0; i < n; i++) {

cin >> c[i];

tot += c[i];

}

for (int i = 0; i < n - 1; i++) {

int x, y;

cin >> x >> y;

x--;

y--;

g[x].push\_back(y);

g[y].push\_back(x);

}

set<ll>\* s = dfs(0);

delete s;

if (res == INF)

res = -1;

cout << res << endl;

}

int main() {

ios\_base::sync\_with\_stdio(0);

int p;

cin >> p;

while (p--) {

solve();

}

return 0;

}

