Travelling Couples

locked

Problem | Submissions | Leaderboard | Discussions

Submitted 6 hours ago • Score: 100.00 Status: Accepted

✔ Test Case #0	✔ Test Case #1	✔ Test Case #2
✔ Test Case #3	✔ Test Case #4	✔ Test Case #5
✔ Test Case #6	✔ Test Case #7	✔ Test Case #8
✔ Test Case #9	✔ Test Case #10	

Submitted Code

Language: C++20 Open in editor

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

// CONSTANTS
const int mod = 1000000007;
// const int mod = 998244353;

// COMMONLY USED TYPES
typedef long long ll;
typedef long double ld;
typedef vector<int> vi;
typedef vector<ll> vll;
typedef pair<int, int> pi;
typedef pair<ll, ll> pl;
typedef vector<pi> vpi;
typedef vector<pl> vpl;

// LOOPS
#define FOR(i, l, r) for (ll i = l; i <= (r); i++)
#define F0R(i, n) for (ll i = 0; i < (n); i++)
#define FORd(i, a, b) for (ll i = (b)-1; i >= a; i--)
#define F0Rd(i, a) for (ll i = (a); i >= 0; i--)
#define trav(x, a) for (auto &x : a)

// SHORT HAND
#define pb push_back
#define mp make_pair
#define fi first
#define se second
#define sz(a) int((a).size())
#define fill(x, y) memset(x, y, sizeof(y))
#define all(x) (x).begin(), (x).end() // Forward traversal
#define endl "\n"
#define max(a, b) (a < b ? b : a)
#define min(a, b) ((a > b) ? b : a)

ll gcd(ll a, ll b)
{
    return b ? gcd(b, a % b) : a;
```

```cpp
}

#define watch(x) cout << (#x) << " is : " << (x) << "\n"
#define watch2(x, y) cout << (#x) << " is " << (x) << " and " << (#y) << " is " << (y) << "\n"

// First Variable Needs to be Ans like ans = max(ans,xyz)
template <class T>
bool ckmin(T &a, const T &b)
{
    return b < a ? a = b, 1 : 0;
}
template <class T>
bool ckmax(T &a, const T &b)
{
    return a < b ? a = b, 1 : 0;
}
//                      BE COOL AND FOCUS ON ACCURATE & SIMPLE CODE!
// ---------------------------------TEMPLATE ENDS~---------------------------------

// All Variables Here

// All Functions Here

// return shortest distance from src to all other nodes vector
vi bfs(int src, int n, vector<vi> &adj)
{
    // source vertex
    int s = src;

    queue<int> q;
    vector<bool> used(n);

    // Distance and Parent Vector
    vector<int> d(n), p(n);

    q.push(s);
    used[s] = true;
    p[s] = -1;
    while (!q.empty())
    {
        int v = q.front();
        q.pop();

        for (int u : adj[v])
        {
            if (!used[u])
            {
                used[u] = true;
                q.push(u);
                d[u] = d[v] + 1;
                p[u] = v;
            }
        }
    }

    // for (int i = 0; i < n; i++)
    // {
    //     cout << "dist[" << i << "] = ";
    //     cout << d[i] << endl;
    // }

    return d;
}

void solve()
{
    int c1, c2, c3;
    cin >> c1 >> c2 >> c3;

    int n; // number of nodes
    int m; // number of edges
    cin >> n >> m;

    vector<vector<int>> adj(n); // adjacency list representation

```

```
    while (m--)
    {
        int x, y;
        cin >> x >> y;
        x--;
        y--;
        adj[x].pb(y);
        // If Bidirectional, then only below line
        adj[y].pb(x);
    }

    vi d1 = bfs(0, n, adj);
    vi d2 = bfs(1, n, adj);
    vi d3 = bfs(n - 1, n, adj);

    ll ans = 1e9;

    for (int i = 0; i < n; i++)
    {
        ll cost = 0;
        // from 0 to point i {Husband}
        cost += (c1 * d1[i]);
        // from 1 to point i {Wife}
        cost += (c2 * d2[i]);
        // from point i to (n-1) or reverse [n-1 to point i]
        cost += (c3 * d3[i]);
        ans = min(ans, cost);
    }

    cout << ans;

    return;
}

int main()
{
    ios_base::sync_with_stdio(false);
    cin.tie(0);
    // If No Test Case, Then Comment it!
    int tc = 1;
    // cin >> tc;
    while (tc--)
    {
        solve();
    }
    return 0;
}
```