Tree Coloring

Submissions Attempted by: **156** | Solved by: **69** | Partially Solved by: **23** | ★★★½☆

Algorithms Combinatorics Dynamic Programming Medium Trees Edit

1 LIVE EVENTS

Since node 1 will always colored first, we can consider the tree is rooted at 1.

Let S[i] be the number of nodes in subtree rooted at node *i*. Let F[i] be the number of ways coloring subtree rooted at node *i*, in which we always colored node *i* first. The answer of this problem is F[1].

How to calculate F[i]? Assuming that node *i* have *k* direct child: c_1, c_2, .., c_k. The order of coloring nodes in each subtree rooted at node c_j are independent to each other. Therefore, we have the formula:

F[i] = C(S[i], S[c_1]) * C(S[i] - S[c_1], S[c2]) * .. * C(S[i] - S[c_1] - S[c_2] - .. - S[c_(k - 1)], S[c_k]) * F[c_1] * F[c_2] * .. *F[c_k], where C(n, k) denoting binomial coefficient (n, k), that means C(n, k) = n! / ((n - k)! * k!).

Edit Editorial

IS THIS EDITORIAL HELPFUL?

Yes, it's helpful

No, it's not helpful

13 developer(s) found this editorial helpful.

Author Solution by Vuong Nguyen

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

using namespace std;

const int MAXN = 100000 + 10;
const int MOD = (int)(1e9) + 7;

vector<int> adj[MAXN];
int fac[MAXN], rev[MAXN], f[MAXN], subtree[MAXN];
int n;

int power(int x, int k, int MOD) {
    if (k == 0) return 1 % MOD;
    long long t = power(x, k / 2, MOD);
    t = (t * t) % MOD;
    if (k % 2 == 1) t = (t * x) % MOD;
    return t;
}

void init() {
    int n = 100000;
    fac[0] = 1;
    for(int i = 1; i <= n; i++) fac[i] = (1LL * i * fac[i - 1]) %
    for(int i = 0; i <= n; i++) rev[i] = power(fac[i], MOD - 2, M
}

int combi(int k, int n) {
    return (1LL * fac[n] * ((1LL * rev[k] * rev[n - k]) % MOD)) %
}

void DFS(int u, int par = -1) {
    f[u] = 1; subtree[u] = 1;
    for(int i = 0; i < adj[u].size(); i++) {
        int v = adj[u][i];
        if (v != par) {
            DFS(v, u);
            subtree[u] += subtree[v];
        }
    }
    int s = subtree[u] - 1;
    for(int i = 0; i < adj[u].size(); i++) {
        int v = adj[u][i];
        if (v != par) {
            int x = (1LL * combi(subtree[v], s) * f[v]) % MOD;
            f[u] = (1LL * f[u] * x) % MOD;
            s -= subtree[v];
        }
    }
}

int main()
{
    init();
    int test;
    cin >> test;
    while (test --) {
        cin >> n;
        for(int i = 1; i <= n; i++) adj[i].clear();
        for(int i = 1; i <= n - 1; i++) {
            int u, v;
            cin >> u >> v;
            adj[u].push_back(v); adj[v].push_back(u);
        }
        DFS(1);
        cout << f[1] << endl;
    }
}
```

Tester Solution by Anta

```
#include <string>
#include <vector>
#include <algorithm>
#include <numeric>
#include <set>
#include <map>
#include <queue>
#include <iostream>
#include <sstream>
#include <cstdio>
#include <cmath>
#include <ctime>
#include <cstring>
#include <cctype>
#include <cassert>
#include <limits>
#include <functional>
#define rep(i,n) for(int (i)=0;(i)<(int)(n);++(i))
#define rer(i,l,u) for(int (i)=(int)(l);(i)<=(int)(u);++(i))
#define reu(i,l,u) for(int (i)=(int)(l);(i)<(int)(u);++(i))
#if defined(_MSC_VER) || __cplusplus > 199711L
#define aut(r,v) auto r = (v)
#else
#define aut(r,v) __typeof(v) r = (v)
#endif
#define each(it,o) for(aut(it, (o).begin()); it != (o).end(); ++ i
#define all(o) (o).begin(), (o).end()
#define pb(x) push_back(x)
#define mp(x,y) make_pair((x),(y))
#define mset(m,v) memset(m,v,sizeof(m))
#define INF 0x3f3f3f3f
#define INFL 0x3f3f3f3f3f3f3f3fLL
using namespace std;
typedef vector<int> vi; typedef pair<int,int> pii; typedef vector<
template<typename T, typename U> inline void amin(T &x, U y) { if(
template<typename T, typename U> inline void amax(T &x, U y) { if(

template<int MOD>
struct ModInt {
        static const int Mod = MOD;
        unsigned x;
        ModInt(): x(0) { }
        ModInt(signed sig) { int sigt = sig % MOD; if(sigt < 0) si
        ModInt(signed long long sig) { int sigt = sig % MOD; if(si
        int get() const { return (int)x; }

        ModInt &operator+=(ModInt that) { if((x += that.x) >= MOD)
        ModInt &operator-=(ModInt that) { if((x += MOD - that.x) >
        ModInt &operator*=(ModInt that) { x = (unsigned long long)
        ModInt &operator/=(ModInt that) { return *this *= that.inv

        ModInt operator+(ModInt that) const { return ModInt(*this)
        ModInt operator-(ModInt that) const { return ModInt(*this)
        ModInt operator*(ModInt that) const { return ModInt(*this)
        ModInt operator/(ModInt that) const { return ModInt(*this)

        ModInt inverse() const {
                long long a = x, b = MOD, u = 1, v = 0;
                while(b) {
                        long long t = a / b;
                        a -= t * b; std::swap(a, b);
                        u -= t * v; std::swap(u, v);
                }
                return ModInt(u);
        }
};
typedef ModInt<1000000007> mint;

vector<mint> fact, factinv;
void nCr_computeFactinv(int N) {
        N = min(N, mint::Mod - 1);
        fact.resize(N+1); factinv.resize(N+1);
        fact[0] = 1;
        rer(i, 1, N) fact[i] = fact[i-1] * i;
        factinv[N] = fact[N].inverse();
        for(int i = N; i >= 1; i --) factinv[i-1] = factinv[i] *
}

vector<int> t_parent;
vi t_ord;

void tree_getorder(const vector<vi> &g, int root) {
        int n = g.size();
        t_parent.assign(n, -1);
        t_ord.clear();

        vector<int> stk; stk.push_back(root);
        while(!stk.empty()) {
                int i = stk.back(); stk.pop_back();
                t_ord.push_back(i);
                for(int j = (int)g[i].size()-1; j >= 0; j --) {
                        int c = g[i][j];
                        if(t_parent[c] == -1 && c != root)
                                stk.push_back(c);
                        else
                                t_parent[i] = c;
                }
        }
}
```

```

int main() {
        int T;
        scanf("%d", &T);
        assert(1 <= T && T <= 10);
        rep(ii, T) {
                int N;
                scanf("%d", &N);
                assert(1 <= N && N <= 100000);
                vector<vi> g(N);
                rep(i, N-1) {
                        int u, v;
                        scanf("%d%d", &u, &v), -- u, -- v;
                        assert(0 <= u && u < N && 0 <= v && v < N
                        g[u].push_back(v);
                        g[v].push_back(u);
                }
                tree_getorder(g, 0);
                vector<int> subtreesize(N, 1);
                for(int ix = N-1; ix > 0; -- ix)
                        subtreesize[t_parent[t_ord[ix]]] += subtree

                nCr_computeFactinv(N);

                vector<mint> dp(N);
                for(int ix = N-1; ix >= 0; -- ix) {
                        int i = t_ord[ix];
                        mint x = 1;
                        int totalsize = 0;
                        each(j, g[i]) if(*j != t_parent[i]) {
                                int size = subtreesize[*j];
                                x *= dp[*j];
                                totalsize += size;
                                x *= factinv[size];
                        }
                        x *= fact[totalsize];
                        dp[i] = x;
                }
                mint ans = dp[0];
                printf("%d\n", ans.get());
        }
        return 0;
}

```

PROFILE IMPACT

Complete Profile

*Excellent profile will increase your profile discoverability and keep you on top among others.

PROBLEMS SUGGESTED FOR YOU

Nth Prime
Solved by 34

Swap these knights!
Solved by 6

Matrix
Solved by 3

more...

RECENT SUBMISSIONS

User	Result	Time	Lang
Shakil A...		2.3989	C++
Shakil A...		2.376	C++
Shakil A...		9.2519	C++
Shakil A...		0.0	C++
vipul sh...		4.1306	C++
Sunil Va...		3.3372	C++
Anarbek ...		25.3736	Java

View All

TRENDING NOTES

Number Theory - III
written by Boris Sokolov

Exact String Matching Algorithms
written by Alei Reyes

Binary Indexed Tree or Fenwick Tree
written by Chandan Mittal

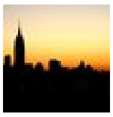
Small tricks in for loop
written by Rangeesh

Strings And String Functions
written by Vinay Singh

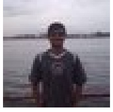
more ...

DEVELOPERS TO FOLLOW

Abhijit
0 followers



Nitesh Singhal
1 followers



Priyank Bhatnagar
45 followers

COMPANIES TO FOLLOW

Medlife International
2058 followers

PERSISTENT
1753 followers

WebEngage
3423 followers

RECOMMENDED CHALLENGES

Horlicks Hack 4 Fun

03 Sep 2015, 09:00 PM IST

Register

CODE-HUNT-2F

21 Oct 2015, 05:00 PM IST

Register

Zoomcar Ruby Challenge

23 Oct 2015, 06:00 PM IST

Register

Zomato Hiring Challenge

23 Oct 2015, 06:00 PM IST

Register

Diona iOS Developer Hiring Challenge

24 Oct 2015, 12:00 PM IST

Register

Tipstat Android Developer Hiring Challenge

24 Oct 2015, 12:00 PM IST

Register

D'code

SUBSCRIBE TO HACKEREARTH NEWS

bhawnesh.dipu@gmail.com

Subscribe

JOIN PROGRAMMING CLUB ON FACEBOOK

Join now

ABOUT US	HACKEREARTH	DEVELOPERS
Blog	API	AMA
Engineering Blog	Chrome Extension	Code Monk

Updates & Releases

Team

Careers

In the Press

CodeTable

HackerEarth Academy

Developer Profile

Resume

Campus Ambassadors

Get Me Hired

Privacy

Terms of Service

Judge Environment

Solution Guide

Problem Setter Guide

Practice Problems

HackerEarth Challenges

College Challenges

RECRUIT

Developer Sourcing

Lateral Hiring

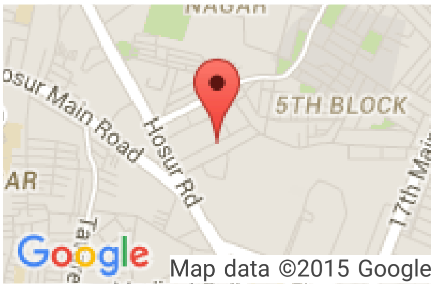
Campus Hiring

FAQs

Customers

Annual Report

REACH US



IIIrd Floor, Salarpuria Business Center,
4th B Cross Road, 5th A Block,
Koramangala Industrial Layout,
Bangalore, Karnataka 560095, India.

✉ contact@hackerearth.com
☎ +91-80-4155-4695
☎ +1-650-461-4192