

```
1 // verified: https://csacademy.com/submission/2149615/
2 #include "xor_shift.hpp"
3 #include "../bits/stdc++.h"
4
5 // 乱数でグラフの同型性判定
6 // Hack 可能.
7 class Isomorphism
8 {
9     using Graph = std::vector<std::vector<int>>>;
10     using ll = long long;
11     const int MOD = 1e9 + 7;
12     int n;
13     XorShift xs;
14     Graph graph;
15     // rnd[i] := 深さ i の頂点の乱数
16     // hash[j] := 頂点 j の乱数
17     std::vector<ll> rnd, hash;
18     ll dfs(int cur, int pre, int dep)
19     {
20         int sz = graph[cur].size();
21         if (sz == 1 && pre != -1)
22         {
23             return hash[cur] = 1;
24         }
25         ll res = 1;
26         for (int i = 0; i < sz; i++)
27         {
28             int next = graph[cur][i];
29             if (next == pre)
30                 continue;
31             res *= (rnd[dep] + dfs(next, cur, dep + 1));
32             res %= MOD;
33         }
34         return hash[cur] = res;
35     }
36
37 public:
38     Isomorphism(int _n) : n(_n), xs(XorShift()), rnd(n, -1), hash(n, -1), graph(n)
39     {
40         for (int i = 0; i < n; i++)
41         {
42             rnd[i] = xs.randInt(0, MOD - 1);
43         }
44     }
45
46     void add_edge(int u, int v)
47     {
48         graph[u].push_back(v);
49     }
50     ll getHash(int root = 0)
51     {
52         return dfs(root, -1, 0);
53     }
54
55     ll getNodeHash(int node = 0)
56     {
57         return hash[node];
58     }
59 };
60
```