



//MOBIUS(X) is zero if X is not a square-free integer.

//MOBIUS(X) is 1 if X is a square-free integer and the number of X's prime divisors is even.

//MOBIUS(X) is -1 if X is a square-free integer and the number of X's prime divisors is odd.

int mobiusMu (Int n)

{

int ans = 1 ;

for (Int d = 2 ; d <= n; ++d)

{

if (n % (d \* d) == 0 )

return 0;

else if (n % d == 0 )

n /= d, ans \*= -1;

}

return ans;

}

for(int i=2; i<=MAXA; i++)

if(!P[i])

{

for(int j=i; j<=MAXA; j+=i)

if(!P[j])

P[j]=i;

}

MOBIUS[1]=1;

for(int i=2; i<=MAXA; i++)

if(P[i]==P[i/P[i]])

MOBIUS[i]=0;

else

MOBIUS[i]=-1 \* MOBIUS[i/P[i]];

BUILD PRUFER CODE

vector<vector<int>> adj;

vector<int> pruefer\_code() {

int n = adj.size();

set<int> leafs;

vector<int> degree(n);

vector<bool> killed(n, false);

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

degree[i] = adj[i].size();

if (degree[i] == 1)

leafs.insert(i);

}

vector<int> code(n - 2);

for (int i = 0; i < n - 2; i++) {

int leaf = \*leafs.begin();

leafs.erase(leafs.begin());

killed[leaf] = true;

int v;

for (int u : adj[leaf]) {

if (!killed[u])

v = u;

}

code[i] = v;

if (--degree[v] == 1)

leafs.insert(v);

}

return code;}

BUILD TREE FROM PRUFER CODE

vector<pair<int, int>> pruefer\_decode(vector<int> const& code) {

int n = code.size() + 2;

vector<int> degree(n, 1);

for (int i : code)

degree[i]++;

set<int> leaves;

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

if (degree[i] == 1)

leaves.insert(i);

}

vector<pair<int, int>> edges;

for (int v : code) {

int leaf = \*leaves.begin();

leaves.erase(leaves.begin());

edges.emplace\_back(leaf, v);

if (--degree[v] == 1)

leaves.insert(v);

}

edges.emplace\_back(\*leaves.begin(), n-1);

return edges;

}