Subgraph Isomorphism Problem

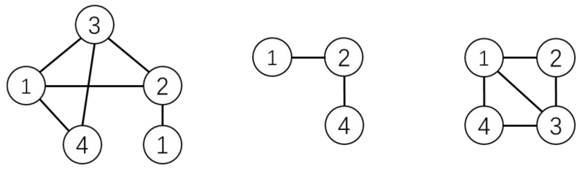
A graph is a data structure consisting of nodes and edges that can represent relationships between entities. Meanwhile, a pattern is a graph that represents a particular structure, such as a cached query or a fraud behavior. The goal of this task is, given the graph G and a number of patterns S_i, to find all the patterns that are present in the graph.

You are given a graph G with a set of nodes V and a set of edges E. The nodes of the graph are conveniently numbered from 1 to n. Each node $v \in V$ has a label L_v. You are also given a set of m patterns S_1, S_2, ..., S_m. Each pattern is also a graph. Pattern S_i has a set of nodes V_i and a set of edges E_i. Nodes of the i-th pattern are conveniently numbered from 1 to n_i. Each node $u \in V_i$ has a label L_u^i for each i from 1 to m.

A pattern $S_i(V_i, E_i)$ is a subgraph isomorphic to the $G(V, E)$ when an injective function $f_i : V_i \to V$ that satisfy the following conditions exists:

- for all nodes $u \in V_i : L_u^i = L_{f(u)}$, which means the mapping node $f(u)$ in graph G has the same label as the node u in S_i;
- for all edges $(x, y) \in E_i : (f(x), f(y)) \in E$, which means there must be an edge between $f(x)$ and $f(y)$ in graph G if there is an edge between x and y in S_i.

Let's consider an example below.



(a) graph $\boldsymbol{G}$ (b) graph pattern $\boldsymbol{S_1}$ (c) graph pattern $\boldsymbol{S_2}$

In this example, the graph G consists of 5 nodes and 6 edges. The numbers on nodes indicate the label of the corresponding node. The graph pattern S_1 is not a subgraph of the graph G because the edge between the node with label 2 and the node with label 4 in S_1 does not exist in the graph G. Graph pattern S_2 is a subgraph of the graph G.

Your task is to check for each of the patterns S_i whether it is a subgraph of the graph G. Please refer to the Scoring section of the statement for more clarifications.MY

Input

The first line of the input contains two integers n and k — the number of vertices and the number of edges in the graph G. The second line contains a list of n space-separated integers L_j — the labels of the vertices of the graph G. The next k lines describe the edges of the graph G. Line number j contains a pair of integers x_j and y_j which are the endpoints of the j-th edge of the graph G.

The next line contains a single integer m — number of patterns.

The rest of the input contains descriptions of the patterns, one by one.

The description of the i-th pattern starts with a line that contains a pair of integers n_i and k_i — the number of vertices and the number of edges in the graph S_i. The second line of the description of the i-th pattern contains a list of n_i space-separated integers L_j^i — the labels of the vertices of the graph S_i. The next k_i lines describe the edges of the graph S_i. Line number j contains a pair of integers x_j^i and y_j^i which are the endpoints of the j-th edge of the graph S_i.

Output

In the first line, print a single integer p — the number of patterns that are subgraphs of the graph G. The i-th of the following p lines should have the following format: ind_i $f_{ind_i}(1)$ $f_{ind_i}(2)$ ... $f_{ind_i}(n_{ind_i})$. This means that the pattern S_{ind_i} is a subgraph of the graph G and $f_{ind_i}(j)$ is the index of the j-th vertex of the graph S_{ind_i} in the graph G.

The values f_{ind_i} should form an injective function $f_{ind_i} : V_{ind_i} \to V$ that satisfies both of the conditions mentioned in the first section of the statement.

If there is an edge between vertices u and v in the pattern number ind_i then there has to be an edge between vertices $f_{ind_i}(u)$ and $f_{ind_i}(v)$ in the graph G.

All ind_i should be different.

Constraints

$1 \le m \le 50\ 000$,
$2 \le n \le 2000$,
$2 \le n_i \le 50$,
$1 \le L_j, L^i_j \le 10$,
$1 \le k \le 12\ 000$,
$1 \le k_i \le 250$,
$1 \le x_j, y_j \le n$, $x_j \ne y_j$,
$1 \le x^i_j, y^i_j \le n_i$, $x^i_j \ne y^i_j$,
the graph G as well as graphs S_i are all connected and don't contain multiple edges.

Samples

Input (*stdin*)	Output (*stdout*)
5 6 4 1 3 2 1 1 2 2 3 3 4 1 3 2 4 4 5 2 3 2 1 2 4 1 2 2 3 4 5 1 2 3 4 1 2 2 3 3 4 1 4 1 3	1 2 2 4 3 1

Notes

The only pattern that is a subgraph of the graph G is S_2. Vertex 1 of the graph S_2 is mapped to vertex 2 of the graph G, vertex 2 is mapped to vertex 4, vertex 3 is mapped to vertex 3, and vertex 4 is mapped to vertex 1. You can check that all edges that are needed for isomorphism are present in the graph G. This sample test case is not a part of neither provisional nor final test sets.

Scoring

Your answer for the test case is considered invalid if your output doesn't match the format specified in the Output section of the statement, or if any of the pattern mappings in your output is invalid. In this case, the score for the test is 0.

If your answer is correct, then the score for the test case is calculated as follows:

$$\left\lfloor \frac{p}{m} \cdot 10^7 \right\rfloor,$$

Where p is the number of pattern subgraphs that you found and m is the total number of patterns in the input.

Please note that you *don't have to check ALL the patterns* S_i *for subgraph isomorphism*, but your score depends on the number of isomorphisms you manage to find. The final score is calculated based on the average score for each test case your answer passed.

Submissions

- The execution time limit is 4 seconds per test case, and the memory limit is 1024 mebibytes.
- The code size limit is 64 kibibytes.
- The compilation time limit is 1 minute.

Quick start

Check the sample solution, which have procedures for reading input data and printing the results. The source code is available for some of the contest programming languages:

- Python
- C++

Tests

All test cases, including provisional and final sets, are generated by the generator. Please note that the generator uses a secure random number generator.