

Permutation Routing

Input file: **standard input**
Output file: **standard output**
Time limit: 1 second
Memory limit: 1024 megabytes

Bobo is given a tree $T = (V, E)$ of n vertices, where there is a number p_i on vertex i initially, and p_1, p_2, \dots, p_n is a permutation of 1 to n , meaning that all integers from 1 to n appear exactly once in p_1, p_2, \dots, p_n .

In each operation, Bobo can select a matching $M \subseteq E$ (M is a matching means that no two edges in M share a common vertex), and for each $(u, v) \in M$, swap the number on vertex u and vertex v (i.e. swap p_u and p_v).

Bobo wants to use at most $3n$ operations to make $p_i = i$ for each $1 \leq i \leq n$, can you please help him?

Input

There are multiple test cases. The first line of the input contains an integer T ($T \geq 1$), indicating the number of test cases. For each test case:

The first line contains a single integer n ($1 \leq n \leq 1000$) — the number of vertices of the tree.

The second line contains n integers p_1, p_2, \dots, p_n ($1 \leq p_i \leq n$, and p is a permutation of 1 to n) — the initial number on vertex i is p_i .

Then follow $n - 1$ lines, each with integers u, v ($1 \leq u, v \leq n, u \neq v$) — meaning that there is an edge between u and v .

It is guaranteed that the sum of n^2 of all test cases will not exceed 10^6 .

Output

For each test case: The first line contains a single integer m ($0 \leq m \leq 3n$) — the number of operations you used.

Then m lines follow, where each line starts with an integer $0 \leq k_i < n$ — denoting the number of edges in the matching you select in the i -th operation. Then k_i integers $t_{i,1}, t_{i,2}, \dots, t_{i,k_i}$ follow, denoting the indexes of edges you select.

Example

standard input	standard output
1	4
5	2 4 3
1 4 2 5 3	1 1
1 2	1 2
2 3	1 4
2 4	
1 5	