
Problem A. Aho-Parasick

Input file: **standard input**
Output file: **standard output**
Time limit: 3 seconds
Memory limit: 512 megabytes

We didn't come up with 5/10 or better joke, but some word mangling should suffice.

The Aho-Corasick algorithm takes a set of strings, which we will denote **the dictionary** as the input. Then it constructs the following structure:

There is a vertex corresponding to every prefix of one or more strings from the dictionary. There is also a vertex corresponding to the empty string. Two trees are formed from this set of vertices.

The first one is called the **trie** and it consists of all edges between nodes corresponding to strings s and t , such that t can be obtained from s by appending a single character.

The second one is called the **suffix links** and it consists of all edges between nodes corresponding to different strings s and t , such that s is a suffix of t and there is no string w different from s and t with a corresponding trie node, such that s is a suffix of w and w is a suffix of t .

It can be shown that both the trie and the suffix links are trees. In this problem all edges are undirected.

In this problem trees are treated as sets of edges, and edges are treated as unordered pairs of vertices.

You are given two trees A and B on the same set of vertices S . Construct a dictionary, such that

1. Applying the Aho-Corasick algorithm to this dictionary yields trees isomorphic to the given ones. In other words, let V denote the set of nodes (vertices) of the constructed trie and suffix links, T — the constructed trie and L — the suffix links. There must exist a bijection $f : S \rightarrow V$, such that $\forall_{c1, c2 \in S} \{c1, c2\} \in A \iff \{f(c1), f(c2)\} \in T, \{c1, c2\} \in B \iff \{f(c1), f(c2)\} \in L$.
2. Total length of all strings in the dictionary doesn't exceed $3 \cdot 10^5$.

The alphabet size is limited by the total length of all strings. The answer is guaranteed to exist.

Note that there are pairs of trees such that it is possible to construct a dictionary which will satisfy the first requirement, but it is not possible to satisfy both. Such test cases are not present in this problem, however, because the answer is guaranteed to exist. In other words, the second requirement is meaningful.

Input

The first line contains a single integer n ($2 \leq n \leq 10^5$), the number of vertices.

Next $n - 1$ lines describe the trie. i -th of them contains two integers a and b , meaning that there is an edge between vertices a and b ($1 \leq a, b \leq n$).

Next $n - 1$ lines describe the suffix links in the same format.

It is guaranteed that both the trie and the suffix links are trees and an answer exists.

Output

The first line should contain a single integer k ($1 \leq k \leq 3 \cdot 10^5$) — the number of the strings.

Next k lines should contain the strings. Each should start with a single integer l ($1 \leq l \leq 3 \cdot 10^5$), length of the string. l integers a_i ($1 \leq a_i \leq 3 \cdot 10^5$) representing the letters of the string should follow. Same a_i correspond to same letters and vice versa.

The total length of all strings should not exceed $3 \cdot 10^5$.

Examples

standard input	standard output
2 1 2 2 1	1 1 228
3 1 2 2 3 3 2 2 1	2 1 1 1 2
3 1 2 2 3 2 1 3 2	1 2 1 1
3 1 2 2 3 3 1 3 2	2 2 1 2 1 1
4 1 2 2 3 4 1 1 4 4 3 2 1	2 2 1 2 1 2
5 1 2 2 3 3 4 4 5 1 2 4 1 3 2 2 5	2 1 2 3 1 2 3
7 1 2 2 3 3 4 1 5 5 6 1 7 1 2 3 5 4 6 6 7 1 5 1 7	3 1 1 2 2 1 3 3 2 1