

New Year's '16 P4 - DIY Christmas Tree

Time limit: 1.0s **Memory limit:** 128M

You have been mailed a disassembled Christmas tree. In the box, there were three things: a set of N branches, a drawing of a [tree](#) with N nodes in total, and a set of safety instructions.

Each branch can be attached to one other branch. The attachment relationship between two branches can be described as follows: if you attach a branch x to another branch y , x is considered a **child** of y . Each branch also has a **distinct** weight, which is represented by an integer from 1 to N .

The safety instructions state:

- Every branch except the lightest branch should be attached to another branch.
- Start by placing the lightest branch at the top of your tree.
- Add all the remaining branches in a predetermined order, one at a time.
- When adding a branch with a weight of q , find the heaviest branch already added to the assembled tree which is strictly lighter than q . Attach the q weight branch to the one found. If the found branch already has children (attached branches), place it to the right side of all of its children.

Before you start, you realize that you must add the branches in a specific order so that the resulting tree will match the diagram's tree.

Input Specification

On the first line, read one integer N . $1 \leq N \leq 50\,000$ holds for all cases, and $N \leq 7$ holds for 15% of the points.

The following N lines depict an adjacency list of the diagram, where the i^{th} line begins with a single integer k denoting the number of children of node i , where $0 \leq k < N$. Following is a space, and k more space separated integers on the same line. Each integer w ($1 \leq w \leq N$) indicates that node w is a child of i .

It is guaranteed that the tree will be rooted at 1.

Please note that for the second section of input, the ordering of children matters. Values of w are given from left to right, and the tree created by your program's output is to match this. Additionally, the values of i should be considered meaningless other than to make reading the input easier. The diagram consists of branches which are unidentified other than by their position relative to other branches.

If you are still confused, please consult the sample input.

Output Specification

Print a permutation of the numbers in the range $[1, N]$, such that the assembled tree matches the one given in the input. We assert that two trees match if, when simultaneously traversing both in a depth-first manner, the number of children contained by the j^{th} branch in one tree is equal to the number of children contained by the j^{th} branch of the other tree for all possible values of j .

If there are multiple possible answers, output any one of them (only one).

If there is no answer, output a single integer 0.

Again, consult the samples for more clarification.

Sample Input 1

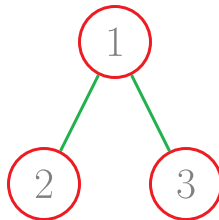
```
3
2 2 3
0
0
```

Sample Output 1

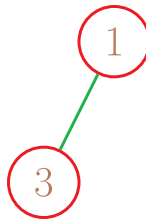
```
1 3 2
```

Explanation for Sample 1

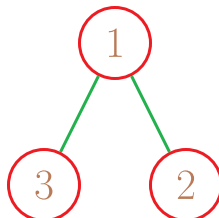
The diagram's tree looks like this:



By starting with the branch with weight 1, then adding branch 3, you get this:

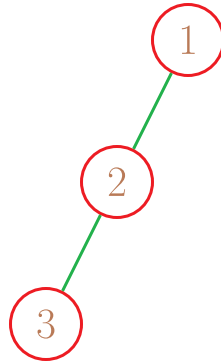


Now add the 2-weight branch:



You can see that the two trees are structurally equivalent. Remember, numbers don't matter.

In this case, the given answer is the only valid answer. For example, adding the branches in the order 1, 2, 3 would produce a different tree:



Sample Input 2

```
7
1 2
2 3 4
1 6
1 7
0
1 5
0
```

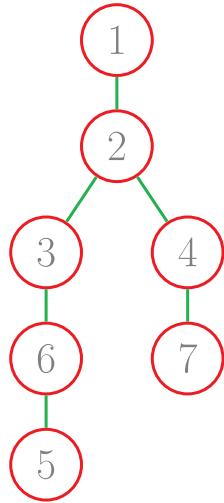
Sample Output 2

```
1 2 5 6 3 7 4
```

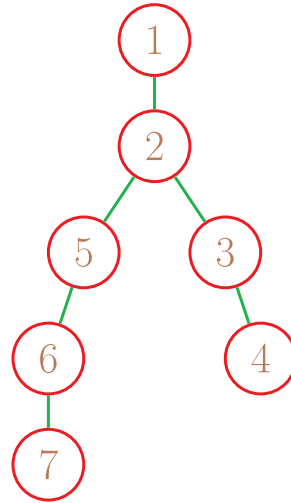
Explanation for Sample 2

The tree looks like this:

On the diagram



After assembling



As you can see, it probably works.