Bitter

Input file:	`standard input`
Output file:	`standard output`
Time limit:	10 seconds
Memory limit:	1024 megabytes

After successfully defrauding his investors, Melon Usk gathered enough money to take over Bitter Inc, and he now has big plans for the company. For a start, Melon wants to reduce Bitter's headcount, and for that, he needs to understand which pairs of employees are *similar*.

The management at Bitter Inc follows a rooted tree structure: employees are numbered from 1 to n, and each employee (apart from Melon i.e. employee 1) has exactly one direct manager, and is either directly or indirectly managed by Mr Usk himself. Employees directly managed by a given employee are *ordered*, and each employee v has an assigned project p_v. Given an employee v, we define their *team* as all employees either directly or indirectly reporting to v (including v themselves). We will say a project p is *owned* by v if all employees working on p are in v's team.

Initially, Melon considered employees x and y similar if their teams could be superimposed onto each other, exactly matching in both management structure and projects p_v that the matched employees work on. However, the resulting number of pairs of similar employees wasn't high enough (considering the number of layoffs Usk is aiming for), so he decided to relax the notion of similarity by allowing *conversions* between projects owned by the compared employees.

Formally, for an employee v, denote their team as T_v, and the projects owned by v as O_v. Consider two employees x and y and a bijection f between O_x and O_y; imagine taking all $v \in T_x$ such that $p_v \in O_x$ and setting p_v to $f(p_v) \in O_y$. If there exists f such that after the above transformation x's team becomes identical* to y's team, then Melon considers x and y similar. Note that if x's and y's teams are identical without any changes and $x \neq y$, then O_x and O_y must be empty, and indeed the employees are similar according to Melon's definition.

If an employee is found to be similar to many other employees, then they may be in for an unpleasant surprise...

You are an intern, recently hired by Melon himself. Write a program that for each employee v computes the number of other employees that are similar to v.

Input

The first line of input contains the number of test cases Z ($1 \leq Z \leq 100\,000$). The descriptions of the test cases follow.

The first line of a test case contains the number of employees n ($1 \leq n \leq 400\,000$).

The second line of a test case contains n numbers p_v ($1 \leq p_v \leq n$) – the project that the respective employee is working on.

The next n lines describe the reporting structure: the i-th line starts with an integer d_i ($0 \leq d_i < n$), denoting the number of i's direct reports; it is then followed by d_i numbers $v_{1,j}$ ($1 \leq v_{1,j} \leq n$), denoting the IDs of the reports. Note that the order of reports does matter.

The total number of employees in all test cases does not exceed 800 000.

Output

For each test case, output a single line containing n numbers c_v – the number of employees (excluding v) that are similar to v.

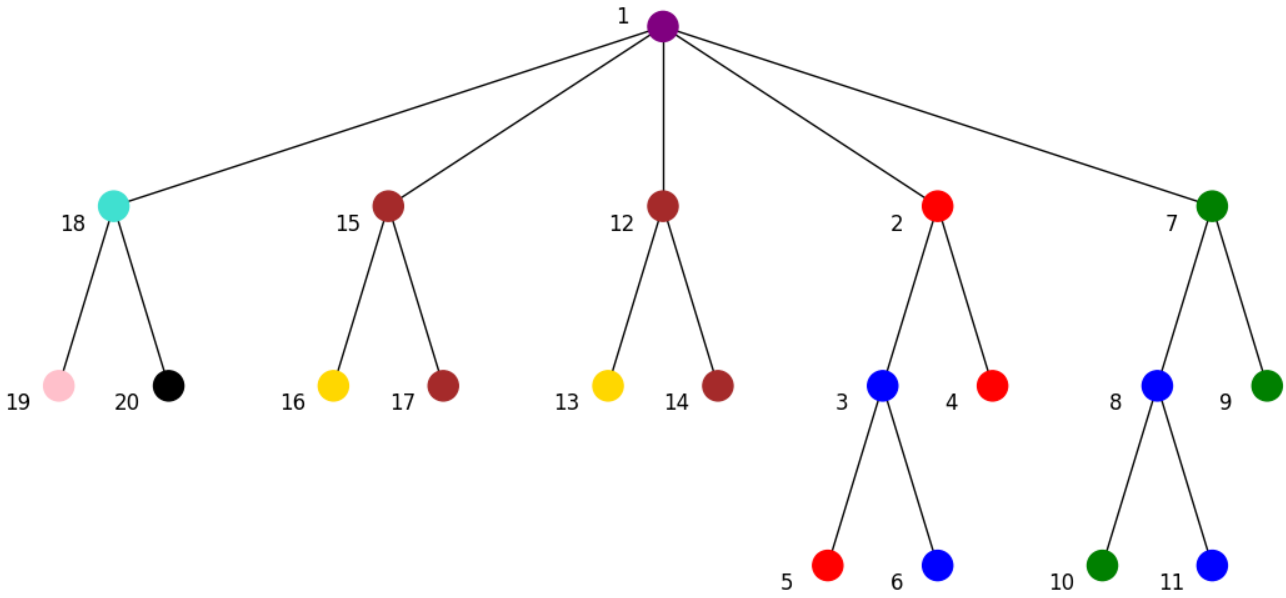
*Formally, two teams are identical if there is a bijection g between them, such that for any node a and an integer k, the k-th child of a is mapped by g to the k-th child of $g(a)$ (note that the order of children matters).

Example

standard input	standard output
1 20 3 1 2 1 1 2 5 2 5 5 2 9 8 9 9 8 9 7 6 4 5 18 15 12 2 7 2 3 4 2 5 6 0 0 0 2 8 9 2 10 11 0 0 0 2 13 14 0 0 2 16 17 0 0 2 19 20 0 0	0 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 0 1 1

Note

The sample test case is depicted below, where different projects are shown as different colors.



The similar pairs of employees are: $(2, 7)$, $(4, 5)$, $(6, 11)$, $(9, 10)$, $(12, 15)$, $(13, 16)$, $(14, 17)$, and $(19, 20)$.

Note that 3 and 8 are not similar, as they differ in one project (red vs green), which are not owned by 3 and 8. However, these projects are owned by 2 and 7, and indeed defining a bijection f to map from red to green shows 2 and 7 are similar. 18 is not similar to any of the employees, which can be inferred from the fact that $|O_{18}| = 3$, and the other employees with the same team structure own at most 1 project (and thus a bijection cannot exist). Employees x and y with no reports are similar to each other either

if x owns p_x and y owns p_y (which is the case for 19 and 20) or if $p_x = p_y$ (which is the case for several other pairs, e.g. 4 and 5).