

1312 – Corrupted Tax Department

There is a strange country, for simplicity we name the country as 'You Know Which', you will know the country after reading the following. The Income Tax department of the country is so corrupted that people use to say that after paying the tax, they are stolen by 'You Know Who'. So, the country is considered as an underdeveloped but leading corrupted countries for a few decades (may be we will see a century or even millennium!). So, people are dissatisfied with the Govt. but still all the taxes paid by general people are taken by 'You Know Who'.

So, some of the talented ACM solvers have realized the fact and they took some initiatives to stop the corruption in the income tax department, and want to save the department from the mysterious 'You Know Who'. In the income tax department, there is an officer who is the head of the department. Every other officer is either subordinate of him or any other officer. But any officer is direct or indirect subordinate of the head. An officer is a subordinate of exactly one other officer.

As the ACM-ers don't know which officers (or the head) are corrupted, so, they planned that every officer or the head should work with his direct subordinate and these two should make a group. The fact is that since they are not in same level, it would be tough for them to join 'You Know Who'. It may be possible that some members cannot form a group, so the ACM-ers want to form maximum number of groups. And they also want to find the number of ways they can form maximum groups.

Input

Input starts with an integer **T** (≤ 30), denoting the number of test cases.

Each case starts with a line containing an integer **n** ($1 \leq n \leq 10000$), where **n** denotes the number of officers (including the head). The head is identified by **1** and others are identified from **2** to **n**. Each of the next **n** lines contains the id of the person, the number of subordinates of this person, and the list of ids of the subordinates of this person separated by spaces. In the list no id is given more than once, and you can assume that the given input follows the restrictions described above.

Output

For each case, print the case number, the maximum number of groups and the number of ways to make maximum groups modulo **10007**.

Sample Input	Output for Sample Input
1 5 1 2 2 4 4 1 5 2 1 3 3 0 5 0	Case 1: 2 3

Note

Dataset is huge, use faster I/O methods.