Problem D. The Hidden Network of Agent Shadow

Time limit 1000 ms
Mem limit 65536 kB

Agent Shadow is the head of a covert spy network. To recruit more agents, Agent Shadow uses a secret strategy. He starts with a certain number of recruitment cards and invites his agents one by one. The process works as follows:

- Agent Shadow (Agent 1) has all the recruitment cards initially. He keeps one card and gives the rest to another agent who has no cards yet, let's say Agent X.
- Agent X, after receiving the cards, keeps one for himself and passes the rest to another agent who hasn't received any cards. The process continues until there are no agents left to invite. At that point, any remaining cards are returned to the agent who invited the current one.
- When an agent finishes distributing the cards, they hand any leftover cards back to the person who invited them. This process is repeated recursively for each agent.

In the end, every agent gets exactly 1 recruitment card. Agent Shadow now needs your help to figure out two things:

1. The total number of invitations that were made.
2. The number of agent pairs that are certainly not connected as friends, meaning they were never directly involved in passing cards.

Input

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

Each case starts with an integer **N ($1 \le N \le 10^5$)**.

Each of the next **N-1** lines will contain two integers, XXX and YYY (indicating that Agent Y was recruited by Agent X **X** and **Y ($1 \le X, Y \le N, X \ne Y$)**.

Output

For each case, print the case number, total number of invitations made, and the number of

different pairs of persons who are surely not friends. See samples for detailed formatting.

Sample

Input	Output
2 2 1 2 3 1 2 1 3	Case 1: 1 0 Case 2: 2 1

Note

Dataset is huge, use faster I/O methods.