Google Code Jam '14 Round 1A Problem B - Full Binary Tree

Time limit: 120.0s **Memory limit:** 1G

A tree is a connected graph with no cycles.

A rooted tree is a tree in which one special vertex is called the root. If there is an edge between X and Y in a rooted tree, we say that Y is a child of X if X is closer to the root than Y (in other words, the shortest path from the root to X is shorter than the shortest path from the root to Y).

A full binary tree is a rooted tree where every node has either exactly 2 children or 0 children.

You are given a tree G with N nodes (numbered from 1 to N). You are allowed to delete some of the nodes. When a node is deleted, the edges connected to the deleted node are also deleted. Your task is to delete as few nodes as possible so that the remaining nodes form a full binary tree for some choice of the root from the remaining nodes.

Input Specification

The first line of the input gives the number of test cases, T. T test cases follow. The first line of each test case contains a single integer N, the number of nodes in the tree. The following $N - 1$ lines each one will contain two space-separated integers: X_i Y_i, indicating that G contains an undirected edge between X_i and Y_i.

Output Specification

For each test case, output one line containing `Case #x: y`, where x is the test case number (starting from 1) and y is the minimum number of nodes to delete from G to make a full binary tree.

Limits

Memory limit: 1 GB.

$1 \leq T \leq 100$.

$1 \leq X_i, Y_i \leq N$.

Each test case will form a valid connected tree.

Small Dataset

Time limit: 60 seconds.

$2 \leq N \leq 15$.

Large Dataset

Time limit: 120 seconds.

$2 \leq N \leq 1000$.

Sample Input

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

Sample Output

```
Case #1: 0
Case #2: 2
Case #3: 1
```

In the first case, G is already a full binary tree (if we consider node 1 as the root), so we don't need to do anything.

In the second case, we may delete nodes 3 and 7; then 2 can be the root of a full binary tree.

In the third case, we may delete node 1; then 3 will become the root of a full binary tree (we could also have deleted node 4; then we could have made 2 the root).

Note

This problem has different time limits for different batches. If you exceed the Time Limit for any batch, the judge will incorrectly display `>120.000s` regardless of the actual time taken. Refer to the **Limits** section for batch-specific time limits.