Task 2. Artillery

You are given a tree with N vertices. There is a pawn at one of the vertices.

You have K artillery cannons and you can make a sequence of moves with your cannons. Each move consists of a shot of all cannons at the vertices chosen by you. So, at each move you hit K vertices of the tree. Between two moves, the pawn can either go to the adjacent vertex of the current one or stay in place. After each move the tree doesn't change. You never know where the pawn is.

Write program **artillery** that finds the minimum value of K with which you can surely hit the pawn.

Input

From the first line of the input, the program reads the number T of the subtests. For each subtest, the value of N is read from a line and each of the following N – 1 lines contains two integers describing an edge in the tree. Each edge is given by its two vertices. The vertices of the tree are numbered starting from 0.

Output

Your program should output T lines. On each line, the program should output the minimum value of K for the corresponding subtest.

Constraints:

$1 \le T \le 10$

$1 \le N \le 100\,000$

In 40% of tests: $N \le 1000$

In other 30% of tests: $N \le 10\,000$

Evaluation

Let a_1, ... a_T be the correct answers for the subtests and b_1, ..., b_T are the corresponding responses received from your program.

Your points for subtest i are:

- 1.0, if $a_i = b_i$
- 0.7, if $a_i = b_i - 1$
- 0 in other cases

Your points for each test are computed as a product of point for all subtests, multiplied by the maximum points that can give this test.

Example

Input

```
2
8
0 1
0 2
0 3
1 4
2 5
3 6
3 7
9
0 1
0 2
0 3
1 4
1 5
1 6
3 7
3 8
```

Output

```
3
2
```

Explanation:

In the first subtest, we can guarantee that we will hit the pawn using 3 canons with the following moves:

1st move: we hit 0, 1 and 4

2nd move: we hit 0, 2 and 5

3rd move: we hit 0, 3 and 6

4th move: we hit 0, 3 and 7

At these moves the pawn will always be hit.

With fewer cannons, the pawn can avoid being hit.