Genealogy

Genealogy is a picture showing the information about the relationship of the members of a family, where the relationship between a parent and a child is represented as the connection by a line.

In genealogy, there is only one person with no parent, who is the progenitor of the family. Then, the progenitor is the 1-level of descendants, and the persons connected by a line with the progenitor, that is, the direct descendants of the progenitor, are the 2-level of descendants. In this way, the person connected by a line with the M-level of descendants is the $(M+1)$-level of descendants in the genealogy.

All the persons in a family are represented as integers and these integers are connected by lines in the genealogy. The relationship that a parent a and a child b are connected by a line in the genealogy is represented as $a\,b$. Then we will write a program to find the maximum M for which there exists at least one M-level of descendants in the genealogy.

For example, the persons in a family are given as 1, ..., 6 and the relationship between a parent and a child is given as follows;

$$3\,5, \quad 1\,2, \quad 4\,1, \quad 3\,4, \quad 4\,6$$

In this genealogy, the progenitor is 3, and there is the maximum 4-level of descendants, who is 2.

[Input]
The first line in the input file has an integer T, the number of the test cases, where T ≤ 45. The number N(2 ≤ N ≤ 50,000) of the persons in a family is given in the first line of each test case. Then, the persons are given as the integers between 1 and N. In each line of the following N-1 lines, two integers a and b (1 ≤ a, b ≤ N) are given to represent the relationship between a parent and a child.

The input is given as one of the following three types.

- Set 1: 2 ≤ N ≤ 10.

- Set 2: 2 ≤ N ≤ 500.

- Set 3: 2 ≤ N ≤ 50,000.

[Output]
For each test case, print the maximum M for which there exists at least one M-level of descendant in the genealogy in a single line.

[I/O Example]
Input

```
2
6
```

```
3 5
1 2
4 1
3 4
4 6
10
5 1
4 5
4 6
6 2
5 8
6 3
7 10
5 9
4 7
```

Output

```
4
3
```