Candy Store

You might now know this about Lea, but besides being a brilliant mathematician she is also an exceptionally gifted confectioner. So far Lea has only been making pralines for her friends, but now she wants to take it a step further and turn her hobby into a career. Lea would like to open a candy store in Lindtown!

The first thing to do is to find a proper location for her store. There are n houses in Lindtown which are connected by $n-1$ bidirectional roads. All roads in Lindtown are of exact same length, and every house in the city can be reached from any other house by traveling along these roads.

Lea believes that she could maximize the turnover if her store was located in a way such that none of her potential customers would have to travel for too long. In other words, she would like to open the store at a location such that the maximal distance between her candy store and any other house in Lindtown is minimized. Lea has already drawn a map of Lindtown but she has trouble finding the perfect location. Maybe you can help her out!

Input

The first line of the input contains an integer t. t test cases follow, each of them separated by a blank line.

Each test case starts with one integer n, where n is the number of houses is Lindtown labeled from 0 to $n-1$. The following $n-1$ lines describe the roads in Lindtown. Each line contains two integers x_i and y_i denoting that the houses labeled with x_i and y_i are connected by a road.

Output

For each test case, output one line containing "Case #i: c" with i being the number of the test case starting at 1, and c being the number of the house at which Lea should open her candy store. If there are multiple optimal locations for Lea's candy store, any of them is acceptable for Lea.

Constraints

- $1 \leq t \leq 10$
- $2 \leq n \leq 10^5$
- $0 \leq x_i, y_i < n$ for all $1 \leq i \leq n$

Sample Input 1

```
3
4
0 1
1 2
2 3

5
0 1
0 2
0 3
0 4

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

Sample Output 1

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

Sample Input 2

```
5
5
4 0
0 3
0 2
4 1

3
1 2
2 0

6
2 5
2 1
1 0
5 3
5 4

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

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

Sample Output 2

```
Case #1: 0
Case #2: 2
Case #3: 2
Case #4: 7
Case #5: 3
```