The airline company NCPC Airways has flights to and from n cities, numbered from 1 to n, around the entire world. However, they only have $n - 1$ different flights (operating in both directions), so in order to travel between any two cities you might have to take several flights. In fact, since the management has made sure that it's possible to travel between any pair of cities, there is exactly one set of flights a passenger have to take in order to travel between two cities (assuming you want to use the same airline).

Recently many of NCPC Airways frequent flyers have complained that they have had to change flights too often to get to their final destination. Since NCPC Airways doesn't want to loose their customers to other airline companies, but still keep the nice property of their flights, they have decided to cancel one of their current flights and replace it with another flight. Help the company by writing a program which finds the best flight to cancel and the best new flight to add so that the maximum number of flight changes a passenger might have to make when travelling between any pair of cities in which NCPC Airways operates is minimized.

The input will be constructed so that it is always possible to improve the maximum number of flight changes needed.

Input

The first line of the input file contains an integer N ($N < 20$) which denotes the total number of test cases. The description of each test case is given below:

The first line of input for each test case contains the integer n ($4 \leq n \leq 2500$), the number of cities NCPC Airways operates in. Then follow $n - 1$ lines specifying the flights. Each flight is given as a pair of cities a and b ($1 \leq a, b \leq n$).

Output

For each test case print three lines of output. The first line should contain an integer, the minimum number of flights needed to take when travelling between any pair of cities after changing one of the flights. The second line should contain two integers, specifying the two cities between which the flight should be canceled. The third line should contain two integers, specifying the two cities where a new flight should be added.

If there are more than one optimal solution, any one of them will be accepted.

Sample Input

```
2
4
1 2
2 3
3 4
14
1 2
1 8
2 3
2 4
8 9
8 10
8 11
4 5
4 6
4 7
10 12
10 13
13 14
```

Sample Output

```
2
3 4
2 4
5
1 8
2 10
```