
Algorithmen und Wahrscheinlichkeit

Programming Exercises

Exercise 1 – *Dining Table*

You decided to organise a dinner party at your home. Since you are an outgoing person your friend pool is very large, but unfortunately not all of your friends are acquainted with each other. In order for the evening to be successful and entertaining for everybody you make the following plan.

For each guest, you write down the list of people they are a friend with. You know that the friendships are a symmetric relation, i.e. that if a person a is a friend of a person b then the person b is a friend of the person a .

Your plan is simple: you want to place each guest on one of the two sides of your table and all of their friends on the other side of the table, so that the conversations are flowing smoothly. You are sure that your table is large enough, but from a first glance at your friendship lists you cannot deduce whether such a placement is possible.

Luckily, you are a fairly good programmer and decide to rely on your programming skills to check if your plan for the evening can happen. Moreover, if your seating plan can work out, you want to know who will be seated on the same side of the table as your best friend.

Input The first line of the input contains the number $t \leq 20$ of test cases. Each of the t test cases is described as follows.

- It starts with a line containing three integers n m r , separated by space, denoting the number of people attending your dinner party ($1 \leq n \leq 10^4$), the total number of friendships ($1 \leq m \leq 10^5$), and the name of your best friend ($0 \leq r \leq n - 1$).
- The next m lines each contain two integers a b , separated by space, indicating that the person a is a friend with the person b (and the other way around, remember—friendships are symmetric!), where $0 \leq a < b \leq n - 1$.

Output For each test case output one line with the people seated on the same side of the table as your best friend r ordered increasingly while respecting your seating plan, or ‘no’ if your seating plan is not possible. If possible, it is guaranteed that this list of people is unique.

Points This exercise is worth 100 points.

Sample Input

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

Sample Output

1 3
0 3 4 5 6
no