Task: ONEWAY
One way roads

The capital of Byteland is suffering from severe traffic congestion. The authorities have decided to make all streets in the city one way to deal with the issue. However, they haven't planned it well enough and there is a risk that after this operation, some parts of the city will become unreachable from others.

Therefore, the transport department prepared a list of pairs of junctions in the city that need to be connected after the operation. Write a program, which will direct all streets in the city, so that all these requirements are satisfied.

Input

The first line of the standard input contains three integers n, m and k ($1 \leqslant n \leqslant 50\,000, 0 \leqslant m, k \leqslant 200\,000$) denoting the number of junctions in the city, the number of streets and the number of requirements respectively. The junctions are numbered from 1 to n. In each of the following m lines there is a pair of integers a_i, b_i ($1 \leqslant a_i, b_i \leqslant n, a_i \neq b_i$) describing one street. Initially all streets are two way. There is at most one street between a pair of junctions. The next k lines describe the requirements. Each of them contains a pair of integers p_i, q_i ($1 \leqslant p_i, q_i \leqslant n, p_i \neq q_i$), meaning that after the streets are made one way, it should be possible to go from junction p_i to q_i.

Output

In the first line of standard output write one word YES or NO, depending on whether the streets can be made one way and fulfill the requirements. If it is possible, m lines should follow, i-th of those lines should contain a pair c_i, d_i which describes how to direct the i-th street from the input. It means that the street should go from c_i to d_i. Note that the order of streets from the input has to be preserved. If there are many possible results, you may output any of them.

For the input data:	the correct result is:
4 4 3	YES
1 2	2 1
2 3	3 2
3 1	1 3
2 4	2 4
1 3	
3 1	
1 4	

Whereas for the input data:	the correct result is:
4 3 2	NO
1 2	
1 3	
1 4	
3 4	
2 3	