Sockets detection

BlurredPixel company is a leading video card manufacturer. The layout of its latest cards consists of many nodes connected with wires transferring the signal that is being processed. There are two types of nodes: processors and sockets. The wire network is mirror-symmetric. In the middle of a card, there is a layer $\mathcal{S}$ of sockets connected to a network $\mathcal{P}_1$ of processors above and to a network $\mathcal{P}_2$ of processors bellow. The processors of network $\mathcal{P}_1$ with their wire connections form a tree having the sockets of layer $\mathcal{S}$ as its only leaves. The processors of network $\mathcal{P}_2$ also form a tree which is the mirror image of the tree formed by network $\mathcal{P}_2$. And again, all the sockets of layer $\mathcal{S}$ are the only leaves of this second tree. Two examples of video cards are given in Image 1.
Recently, the company found notes on a breakthrough video card layout, made by their former chief engineer who mysteriously disappeared. Unfortunately, the notes are not complete. There is a flash disk included which contains a data file with all connections of the layout, but it is not clear which nodes are processors and which are sockets. The company wonders if the data file can be used to reconstruct the layout.

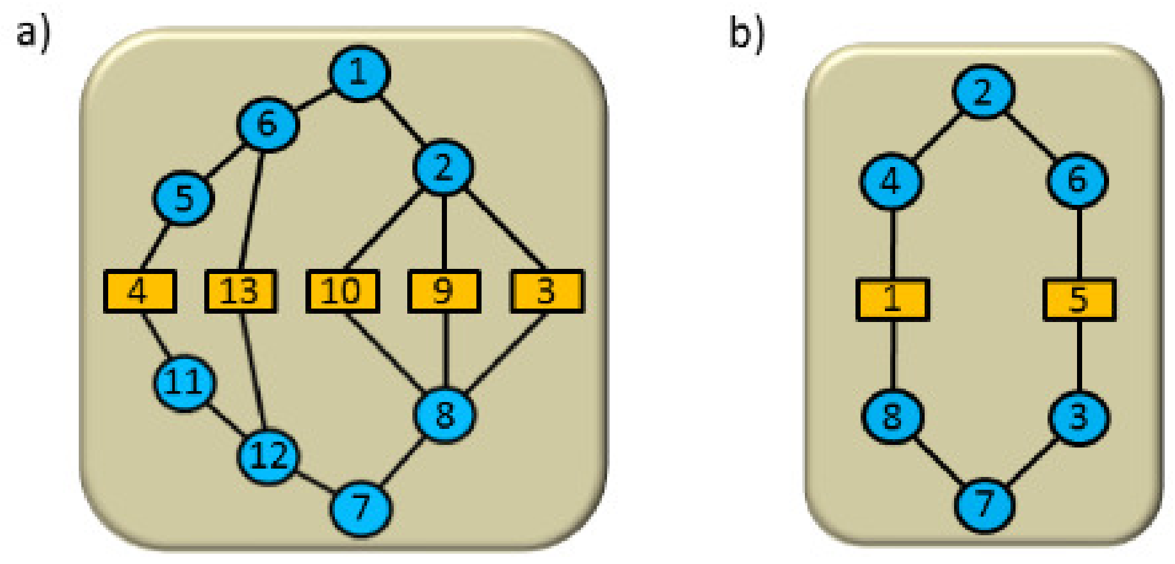

Image 1. Examples of video card layouts. Processors are depicted as blue circles, sockets are depicted as orange rectangles. Wires connecting nodes are represented by black lines.

The task

Given a list of nodes and their wire connections, your task is to detect sockets.

Input

The first input line contains two integers N and M separated by a space. N is the number of nodes, M is the number of connections. Next, there are M lines, each of them containing integers A and B, separated by a space. Each such pair represents a connection between nodes A and B. Nodes are numbered from 1 to N. The nodes numbering as well as the order of connections is arbitrary. It holds that $N \leq 320000$ and $M \leq 400000$.

Output

The output consists of one line containing up to 100 integers, separated by spaces. The integers are the node numbers of the first 100 sockets in ascending order. If there are only $S < 100$ sockets, then only numbers of these S sockets are listed (in ascending order). If there are multiple feasible solutions to an input instance, the lexicographically minimal sequence of socket numbers is expected as the output.

Example 1

Input

```
13 16
1 6
5 6
5 4
13 6
7 12
11 12
11 4
13 12
7 8
2 1
8 10
9 2
2 10
2 3
3 8
9 8
```

Output

```
3 4 9 10 13
```

The input of Example 1 data are depicted in Image 1a).

Example 2

Input

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

Output

```
1 5
```

The input data of Example 2 are depicted in Image 1b).
Remark: There are more feasible solutions for the given input. Except the nodes 1 and 5, the sockets can also be e.g. the nodes 3 and 4. Since sequence 1 5 is the lexicographically minimal feasible solution, it is printed as the output.

Example 3

Input

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

Output

```
4 7 13
```

Public data

The public data set is intended for easier debugging and approximate program correctness checking. The public data set is stored also in the upload system and each time a student submits a solution it is run on the