4768 Mission Impossible

A directed graph with input and output (referred as DGIO hereafter) is a graph in which some nodes have dangling edges, such as the one shown in Figure 1, in which the dangling inputs are a, b, c, d, and the dangling outputs are e, f. A DGIO is "sound" if every dangling input is connected with every dangling output through a directed path. The DGIO shown in Figure 1, for example, is unsound, as input c is not connected with output f, and input d is not connected with output e.

Given a DGIO, your first step is to determine whether it is sound. (easy, isn't it? keep reading) If it is unsound, ideally we would like to split it into a minimum number of sound DGIOs. Figures 2 and 3 show two ways to split the unsound DGIO in Figure 1. Figure 3 is a better split than Figure 2 as it splits Figure 1 into fewer DGIOs. Note, that after a split, the edges connecting different DGIOs become new dangling inputs / outputs.

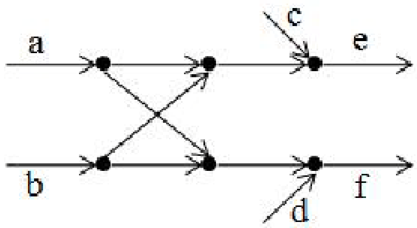

Fig. 1 Unsound DGIO

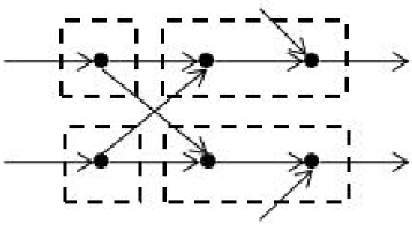

Fig. 2 Split into 4 DGIOs

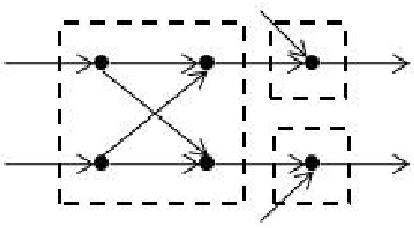

Fig. 3 Split into 3 DGIOs

Computing the minimum split of an unsound DGIO sounds tricky, isn't it? You know what, it is NP-hard... We understand that you folks are extremely good at searching and pruning for NP-hard problems, but no matter how skillful you are, an NP-hard problem IS an NP-hard problem, and an exponential algorithm is never going to be scalable and is prohibitively costly on large-scale graphs.

You are, instead, required to compute a "local-optimal" split for an unsound DGIO. A split is local optimal, if no output DGIOs can be merged to form a sound DGIO. As an example, the split in Figure 4 is NOT a local-optimal split of the DGIO in Figure 1, the four DGIOS on the left hand side in Figure 4 can be merged into a sound DGIO, i.e., resulting in Figure 3. On the other hand, both Figures 2 and 3 are local-optimal splits.

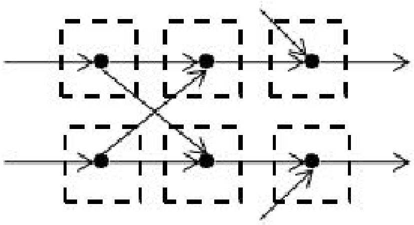

Fig. 4 A Split that is NOT Local-optimal

Input

Each test case begins with the number of nodes N ($2 \leq N \leq 10000$) and edges M. Nodes are numbered from 1 to N. Each of the next M lines containing 2 integers u and v, describing a directed edge from u to v. The edge is a dangling input / output if $u = 0$ / $v = 0$.

Each node has at least one incoming edge and one outgoing edge.

Output

For each test case, if it is sound, simply print '`Sound`'. Otherwise, first print the number of output DGIOs. For each output DGIO, output the number of nodes it comprises, followed by the node IDs.

The order of the DGIOs and node IDs within a DGIO in your output can be arbitrary. Every two words/numbers in your output should be separated by space(s) or EOL(s). No other characters are allowed. Test case 2 of sample input corresponds to Fig. 1, and the output corresponds to Fig. 2.

Sample Input

```
2 3
0 1
1 2
2 0
6 12
0 1
1 2
2 3
3 0
0 3
0 4
4 5
5 6
6 0
0 6
1 5
4 2
```

Sample Output

```
Sound

4
1
1
2
2 3
1
4
2
5 6
```