Problem K
Kernel Scheduler



You are developing the scheduling module for the new operating system. This module takes n tasks to be executed and the dependencies between them and then puts them in a certain order for execution.

More formally, there are n tasks numbered from 1 to n. You are also given m dependencies numbered from 1 to m; i-th of them is described by two numbers — a_i and b_i, meaning that the task a_i should be executed before the task b_i.

In some cases, there are *cyclical dependencies* — situations when according to the dependencies given some task t_1 should be executed before t_2, t_2 before t_3, ..., and t_{k-1} before t_k and t_k before t_1. Cyclical dependencies create a problem for scheduling, so you decided to remove some of the given dependencies in such a way that the resulting set does not contain any cyclical ones.

However, you still need to keep at least $m/2$ original dependencies to preserve some of the original information. You are to write the program performing this task.

Input

- One line containing the numbers n and m ($2 \leq n \leq 10^5$, $1 \leq m \leq 3 \cdot 10^5$).
- m further lines, each containing two numbers a_i and b_i ($1 \leq a_i, b_i \leq n$, $a_i \neq b_i$), describing the corresponding dependency between two tasks a_i and b_i.

Output

The first line should should contain `YES` in case the desired subset of dependencies exists, and `NO` otherwise.

In the `YES` case second line should contain the number k of the selected dependencies (please note that k should be at least $m/2$) and the third line should contain k numbers — the ids of the selected dependencies. They are numbered from 1 to m in the order given in the input.

Sample Input 1	Sample Output 1
3 3 1 2 2 3 3 1	YES 2 1 2

Sample Input 2

```
2 5
1 2
1 2
1 2
2 1
2 1
```

Sample Output 2

```
YES
3
1 2 3
```

Sample Input 3

```
4 4
1 2
2 3
2 4
3 4
```

Sample Output 3

```
YES
4
1 2 3 4
```