CEOI 2021

CENTRAL EUROPEAN OLYMPIAD IN INFORMATICS 2021

ZAGREB, CROATIA | SEPTEMBER 1-5

Day 1

September 2nd 2021

Tasks

Task	Time Limit	Memory Limit	Score
Diversity	7 seconds	512 MiB	100
L-triominoes	8 seconds	512 MiB	100
Newspapers	1 second	512 MiB	100
Total			300

REPUBLIC OF CROATIA
Ministry of Science and Education

CROATIAN ASSOCIATION OF TECHNICAL CULTURE

CROATIAN COMPUTER SCIENCE ASSOCIATION

Task: Diversity

Zoran is a Zookeeper at the Zagreb Zoo. He is currently conducting scientific research on the relationship between visitor satisfaction and the way animals are presented throughout the zoo.

A visitor's walk through the entire zoo can be viewed as a sequence of N habitats each containing a certain species of animal. The i-th habitat initially contains animals of species a_i, and visitors are expected to observe the habitats in order.

Zoran started experimenting with the order in which the animals were presented. He then proceeded to ask questions about visitor satisfaction during their visit. It turned out that the visitors are most satisfied when the *total diversity* of their walk is as small as possible.

Here, the *diversity* of a sequence of habitats is defined as the number of different animal species one can observe in these habitats, whereas the *total diversity* of a sequence of habitats is defined as the sum of diversities over all of its contiguous subsequences.

For example, the diversity of a sequence $(1, 1, 2)$ is 2 because there are two distinct animal species in that sequence. Meanwhile, the total diversity of that sequence is 8 because its contiguous subsequences (1), (1), (2), $(1, 1)$, $(1, 2)$ and $(1, 1, 2)$ have diversities equal to 1, 1, 1, 1, 2 and 2 respectively.

Zoran knows which animal species currently occupies which habitat. Before he reorganizes the zoo, he would like you to answer Q **independent** queries. In the i-th query, he wants to know the **smallest** possible **total diversity** of the contiguous subsequence starting from the l_i-th and ending at the r_i-th habitat that he can achieve by **reordering** the animals living in these habitats.

Input

The first line contains two integers N and Q from the task description.

The second line contains N space-separated integers a_1, a_2, …, a_N. The i-th of these integers represents the animal species occupying the i-th habitat.

Each of the next Q lines contains two integers l_i and r_i ($1 \leq l_i \leq r_i \leq N$) describing a single query.

Note that all queries are independent of each other, i.e. you should answer them under the assumption that the i-th habitat is initially occupied by the animal species a_i.

Output

The i-th line of output should contain a single integer representing the answer to the i-th query.

Scoring

Subtask	Score	Constraints
1	4	$1 \leq N \leq 11$, $1 \leq a_i \leq 300\,000$, $Q = 1$, $l_1 = 1$, $r_1 = N$
2	10	$1 \leq N \leq 300\,000$, $1 \leq a_i \leq 11$, $Q = 1$, $l_1 = 1$, $r_1 = N$
3	8	$1 \leq N \leq 300\,000$, $1 \leq a_i \leq 23$, $Q = 1$, $l_1 = 1$, $r_1 = N$
4	16	$1 \leq N \leq 300\,000$, $1 \leq a_i \leq 1\,000$, $Q = 1$, $l_1 = 1$, $r_1 = N$
5	26	$1 \leq N \leq 300\,000$, $1 \leq a_i \leq 300\,000$, $Q = 1$, $l_1 = 1$, $r_1 = N$
6	36	$1 \leq N \leq 300\,000$, $1 \leq a_i \leq 300\,000$, $1 \leq Q \leq 50\,000$

Examples

input

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

output

```
10
```

input

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

output

```
3
6
```

input

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

output

```
16
8
4
```

Clarification of the first example: In any ordering, the diversity of every contiguous subsequence is equal to the number of its elements. Hence, the answer to the query is $1 + 1 + 1 + 2 + 2 + 3 = 10$.

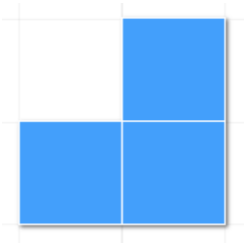
Clarification of the second example: In any ordering, every contiguous subsequence has diversity equal to 1. Thus, for each query, the answer is the number of contiguous subsequences contained in the given contiguous subsequence.

Clarification of the third example: In the first query, an optimal ordering is $(1, 2, 2, 3)$, which has total diversity equal to $1+1+1+1+2+1+2+2+2+3 = 16$. In the second query, an optimal ordering is $(1, 1, 2)$, which has total diversity equal to $1 + 1 + 1 + 1 + 2 + 2 = 8$. In the third query, an optimal ordering is $(1, 3)$, which has total diversity equal to $1 + 1 + 2 = 4$.

Task: L-triominoes

Luka stumbled upon a rectangular board of height H and width W divided into $W \times H$ unit squares. He quickly noticed that exactly K of those unit squares are missing.

Interestingly enough, Luka just happens to have an infinite supply of *L-shaped triominoes*. Is it possible to tile the given board using these triominoes?



We consider the board to be correctly tiled if each unit square of the board is covered by a triomino square. Additionally, triominoes must not cover any of the missing squares, and should not overlap or stick out of the board. Of course, triominoes can be arbitrarily rotated by multiples of 90 degrees.

Input

The first line contains three integers W, H and K ($0 \leq K \leq W \cdot H$) from the task description.

The i-th of the next K lines contains two integers x_i ($1 \leq x_i \leq W$) and y_i ($1 \leq y_i \leq H$), representing the coordinates of the i-th missing square. The given missing squares are pairwise distinct.

Output

If Luka can successfully tile the given board, output `"YES"` in a single line. Otherwise, output `"NO"` in a single line.

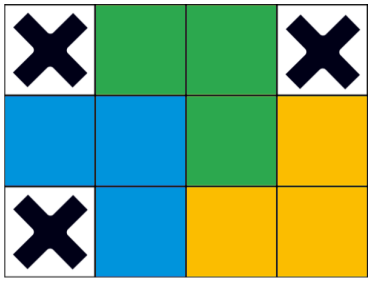
Scoring

Subtask	Score	Constraints
1	10	$2 \leq W \leq 13, 2 \leq H \leq 1\,000, K \leq 250$
2	7	$2 \leq W \leq 13, 2 \leq H \leq 10^9, K = 0$
3	11	$2 \leq W \leq 3, 2 \leq H \leq 10^9, K \leq 250$
4	17	$4 \leq W \leq 6, 2 \leq H \leq 10^9, K \leq 250$
5	35	$7 \leq W \leq 13, 2 \leq H \leq 10^9, K \leq 250$
6	20	$2 \leq W \leq 13, 2 \leq H \leq 10^9, K \leq 250$

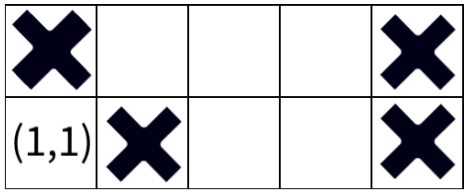
Examples

input	input	input
4 3 3 1 1 1 3 4 3	5 2 4 1 2 2 1 5 1 5 2	2 3 0
output	**output**	**output**
YES	NO	YES

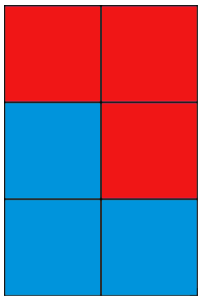
Clarification of the first example:



Clarification of the second example: Luka cannot place a valid triomino which tiles square $(1, 1)$.



Clarification of the third example:



Task: Newspapers

„*Ulovi me, ulovi me, kupit ću ti novine!*" – a popular play song among Croatian children. Translates to *catch me, and I'll buy you newspapers.*

Ankica and Branko are playing a chasing game on an undirected, connected graph. Namely, Branko is moving around the graph, while Ankica is attempting to catch him. The game proceeds in turns, and a single turn consists of the following:

- **Ankica makes a guess on Branko's whereabouts.** More precisely, she guesses that Branko is currently located at a specific node. If she guesses correctly, Branko is caught and the game ends. Otherwise,
- **Branko traverses an edge incident to his current location.** In other words, Branko moves to one of his neighbouring nodes. Note that Branko cannot stay at his present location.

Given a graph, determine if Ankica has a finite strategy which always catches Branko regardless of the way Branko plays and what his starting position may be.

More formally, we represent Ankica's strategy as an array $A = (a_1, a_2, \ldots, a_k)$, where a_i denotes Ankica's guess in the i-th turn (i.e. she guesses that Branko is located in the node a_i).

Similarly, we represent Branko's movements as an array $B = (b_1, b_2, \ldots, b_k)$, where b_i represents the node in which Branko is located before the i-th turn. Additionally, for each two successive elements b_i and b_{i+1} $(1 \leq i < k)$, there must exist an edge in the graph connecting nodes b_i and b_{i+1}. Note that no such constraint is imposed on array A.

We say that Ankica's strategy is successful, i.e. she catches Branko in at most k turns, if, for every valid array B of length k, there exists some i $(1 \leq i \leq k)$ such that $a_i = b_i$ holds.

If such strategy exists, you should find one that minimizes the number k.

You can score some points in this task if you are able to provide a succesful, but not optimal, strategy for Ankica (i.e. a strategy where k is not minimal). See the *Scoring* section for more details.

Input

The first line contains two integers N and M $(N - 1 \leq M \leq \frac{N(N-1)}{2})$ that represent the number of nodes and edges in the graph (respectively). Nodes of the graph are denoted with integers from 1 to N.

The i-th of the next M lines contains two space-separated integers u_i and v_i $(1 \leq u_i, v_i \leq N, u_i \neq v_i)$, representing that an undirected edge connects nodes u_i and v_i.

No edge will appear more than once in the input, and the graph will be connected.

Output

If there is no successful strategy for Ankica, simply output `"NO"` in the first line and terminate the program.

Otherwise, you should output `"YES"` in the first line.

The second line should contain the number k from the task description.

The third line should contain k numbers a_1, a_2, …, a_k from the task description.

Scoring

Subtask	Score	Constraints
1	12	$1 \leq N \leq 20$
2	8	$1 \leq N \leq 1\,000$, $M = N - 1$, each node $u = 1, \ldots, N-1$ is connected to node $u+1$
3	80	$1 \leq N \leq 1\,000$

If your program correctly outputs `"YES"` on the first line on a specific test case, but fails to provide a successful strategy, that test case will be scored with 50% of the points allocated for the subtask it is a part of.

If your program correctly outputs `"YES"` on the first line and provides a successful, but not optimal strategy, that test case will be scored with 75% of the points allocated for the subtask it belongs to. Additionally, in order to score these points, the number of turns (k) in the outputted strategy must be at most $5N$. It can be proved that the number of turns in the optimal strategy doesn't exceed $5N$.

The score for each subtask equals to the smallest score obtained by one of its test cases.

Examples

input

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

output

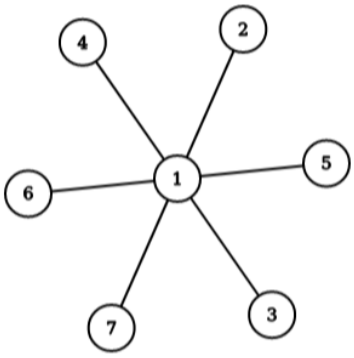
```
YES
2
1 1
```

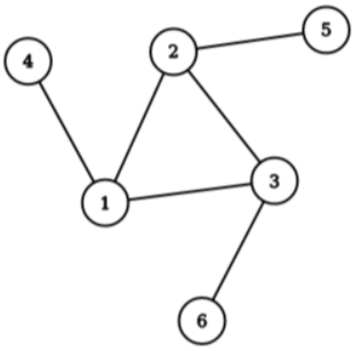
input

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

output

```
NO
```





Clarification of the first example: If Branko is initially located at node 1, he will be caught on the first turn. Otherwise, he will be caught on the second turn.

Clarification of the second example: Let Branko's initial location be among nodes 1, 2 or 3, and different from a_1. Each of these nodes is connected to the other two, so Branko has two options to move to during each turn. At least one option must be safe, so Ankica has no successful strategy.