

# The Contest Contest 1 P6 - A Typical DMOPC Problem

---

**Time limit:** 3.0s    **Memory limit:** 256M  
Java: 5.0s  
Python: 5.0s

---

Alice challenges Bob to a puzzle! Alice is situated on an undirected graph consisting of  $N$  nodes and  $M$  distinctly-weighted edges, the  $i^{\text{th}}$  of which connects node  $a_i$  and  $b_i$ . She starts walking from node 1 and performs the following procedure:

Suppose Alice is at node  $n$ . If  $n = N$ , she stops. Otherwise, she considers the set  $S$  of all edges connected to node  $n$ , excluding the edge she just travelled on, if any. If  $S$  is empty, she stops. Otherwise, she can choose to walk along the lowest or highest weight edge in  $S$ . Note that all edge weights are distinct so there are no ties.

Alice claims that no matter how the edges are weighted, she always has a strategy to never reach node  $N$ . That is, she can choose edges to walk along such that she either stops at some node  $n \neq N$  or ends up in an infinite loop. Skeptical of her claim, Bob gets to work trying to find counterexamples. Can you help Bob find an assignment of edge weights that disproves her claim, or confirm that her claim is correct?

## Constraints

---

$$2 \leq N \leq 5 \times 10^5$$

$$1 \leq M \leq 5 \times 10^5$$

$$a_i \neq b_i$$

### Subtask 1 [10%]

The graph is a tree.

### Subtask 2 [20%]

Every node has degree at least 3.

### Subtask 3 [70%]

No additional constraints.

## Input Specification

---

The first line contains two integers  $N$  and  $M$ .

The next  $M$  lines each contain two integers  $a_i$  and  $b_i$ , indicating that the  $i^{\text{th}}$  edge connects nodes  $a_i$  and  $b_i$ .

## Output Specification

---

On the first line, output either `YES` or `NO`, indicating whether or not there exists an assignment of edge weights that forces Alice to always reach node  $N$ .

If such an assignment exists, output  $M$  lines, where the  $i^{\text{th}}$  line is the weight you assign to the  $i^{\text{th}}$  edge. All weights must be distinct integers between 1 and  $10^9$  (inclusive).

## Scoring

---

For each test file:

If you do not correctly determine if an assignment of edge weights exists, you will receive a score of 0 for that file and a `Wrong Answer` verdict.

Otherwise, on each test case where an assignment is possible, your score for the test file is determined as follows:

You will receive full points if you output a valid assignment of edge weights that is fully correct and 50% of the points otherwise.

Your final score within a subtask will be the minimum score over all tests in that subtask.

## Sample Input 1

---

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

## Sample Output 1

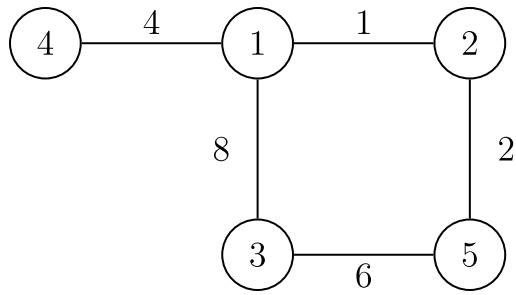
---

```
YES
1
8
4
2
6
```

## Explanation for Sample 1

---

We can weight the edges to obtain the following graph:



Starting from node 1, Alice could either travel to node 2 (using the minimum edge) or node 3 (using the maximum edge) . If she chooses to go to node 2, her only next option is to take the edge to node  $N$ . The same thing would happen if she chooses to go to node 3. Therefore, Alice must always reach node  $N$ .

## Sample Input 2

```

5 7
1 2
2 3
3 4
4 2
4 2
2 5
5 1

```

## Sample Output 2

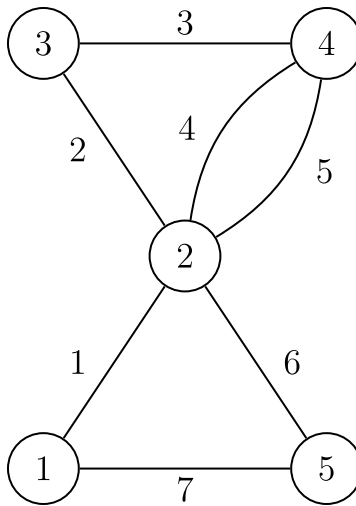
```

YES
1
2
3
4
5
6
7

```

## Explanation for Sample 2

We can weight the edges to obtain the following graph:



It can be shown that Alice must always reach node  $N$ .

### Sample Input 3

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

### Sample Output 3

```
NO
```

### Explanation for Sample 3

It can be proven that no matter how we weight the edges, there exists a strategy for Alice to never reach node  $N$ .