

Bounded Spanning Tree

You are given a connected undirected edge-weighted graph with n vertices and m edges. There are no self-loops in this graph (that is, there is no edge which goes from a vertex to itself), but there can be multiple edges between some pairs of vertices.

Your friend told you the following about this graph:

- The edge weights are **distinct** integers from the range $[1, m]$. In other words, they form some permutation of integers from 1 to m .
- The weight of the i -th edge is from the range $[l_i, r_i]$ for each i from 1 to m .
- The edges with indices $1, 2, \dots, n - 1$ (the first $n - 1$ edges in the input) form a **minimum** spanning tree of this graph.

You want to know if it is possible. Determine if there exist such assignments of edge weights for which these conditions hold and if yes, find any of them.

As a reminder, a spanning tree of a graph is any subset of its edges that forms a tree (connected graph on n vertices with $n - 1$ edges). The minimum spanning tree of a graph is any spanning tree with the smallest sum of weights among all spanning trees of the graph.

Input

The first line contains a single integer t ($1 \leq t \leq 10^5$) - the number of test cases. The description of test cases follows.

The first line of each test case contains two integers n and m ($1 \leq n - 1 \leq m \leq 5 \cdot 10^5$) - the number of vertices and the number of edges, respectively.

The i -th of the following m lines contains four integers u_i, v_i, l_i, r_i ($1 \leq u_i < v_i \leq n$, $1 \leq l_i \leq r_i \leq m$) - indicating that there is an edge connecting vertices u_i, v_i , and that its weight should be in range $[l_i, r_i]$.

It's guaranteed that for each test case, edges with indices $1, 2, \dots, n - 1$ form a spanning tree of the given graph.

It's guaranteed the sum of m over all test cases doesn't exceed $5 \cdot 10^5$.

Output

For each test case, if an array of edge weights that satisfy the conditions doesn't exist, output "NO" in the first line.

Otherwise, in the first line, output "YES". In the second line output m integers w_1, w_2, \dots, w_m ($1 \leq w_i \leq m$, all w_i are **distinct**) - the edge weights (where w_i is the weight assigned to the i -th edge in the input).

If there are multiple answers, output any of them.

You can output each letter in any case (for example, "YES", "Yes", "yes", "yEs", "yEs" will be recognized as a positive answer).

Example

Input:

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

Output:

```
YES
2 3 1 5 4 6
NO
YES
1 2 3 6 4 5
```

Scoring

1. (4 points): $l_i = r_i$ ($1 \leq i \leq m$)
2. (6 points): The sum of m over all test cases doesn't exceed 10
3. (10 points): The sum of m over all test cases doesn't exceed 20
4. (10 points): $m = n - 1$, the sum of m over all test cases doesn't exceed 500
5. (7 points): $m = n - 1$
6. (20 points): $m = n$
7. (11 points): The sum of m over all test cases doesn't exceed 5000
8. (8 points): $u_i = i, v_i = i + 1$ ($1 \leq i \leq n - 1$)
9. (12 points): The sum of m over all test cases doesn't exceed 10^5
10. (12 points): No additional constraints