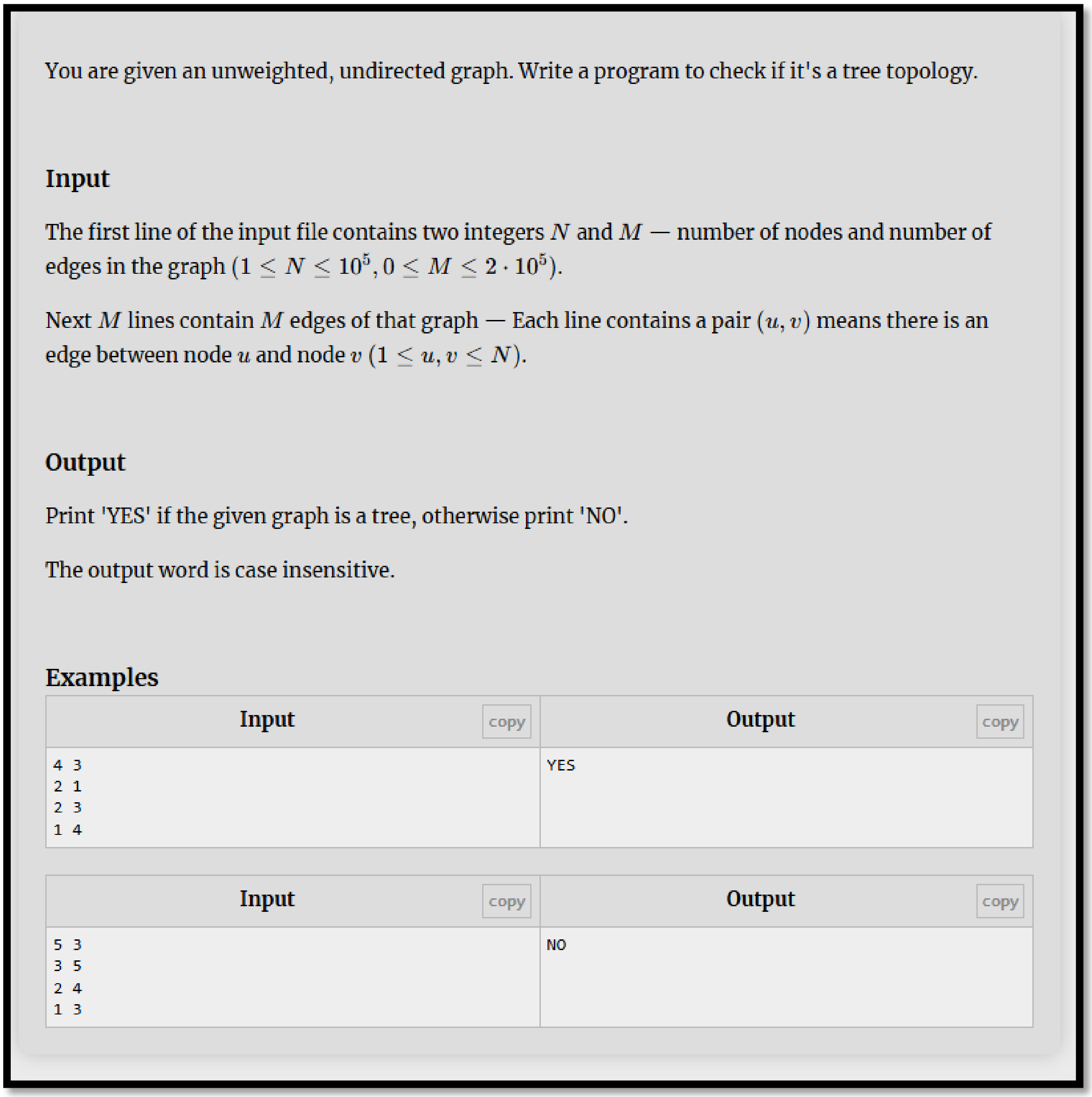
Quiz3

Q1:



You are given an unweighted, undirected graph. Write a program to check if it's a tree topology.

Input

The first line of the input file contains two integers N and M — number of nodes and number of edges in the graph $(1 \leq N \leq 10^5, 0 \leq M \leq 2 \cdot 10^5)$.

Next M lines contain M edges of that graph — Each line contains a pair (u, v) means there is an edge between node u and node v $(1 \leq u, v \leq N)$.

Output

Print 'YES' if the given graph is a tree, otherwise print 'NO'.

The output word is case insensitive.

Examples

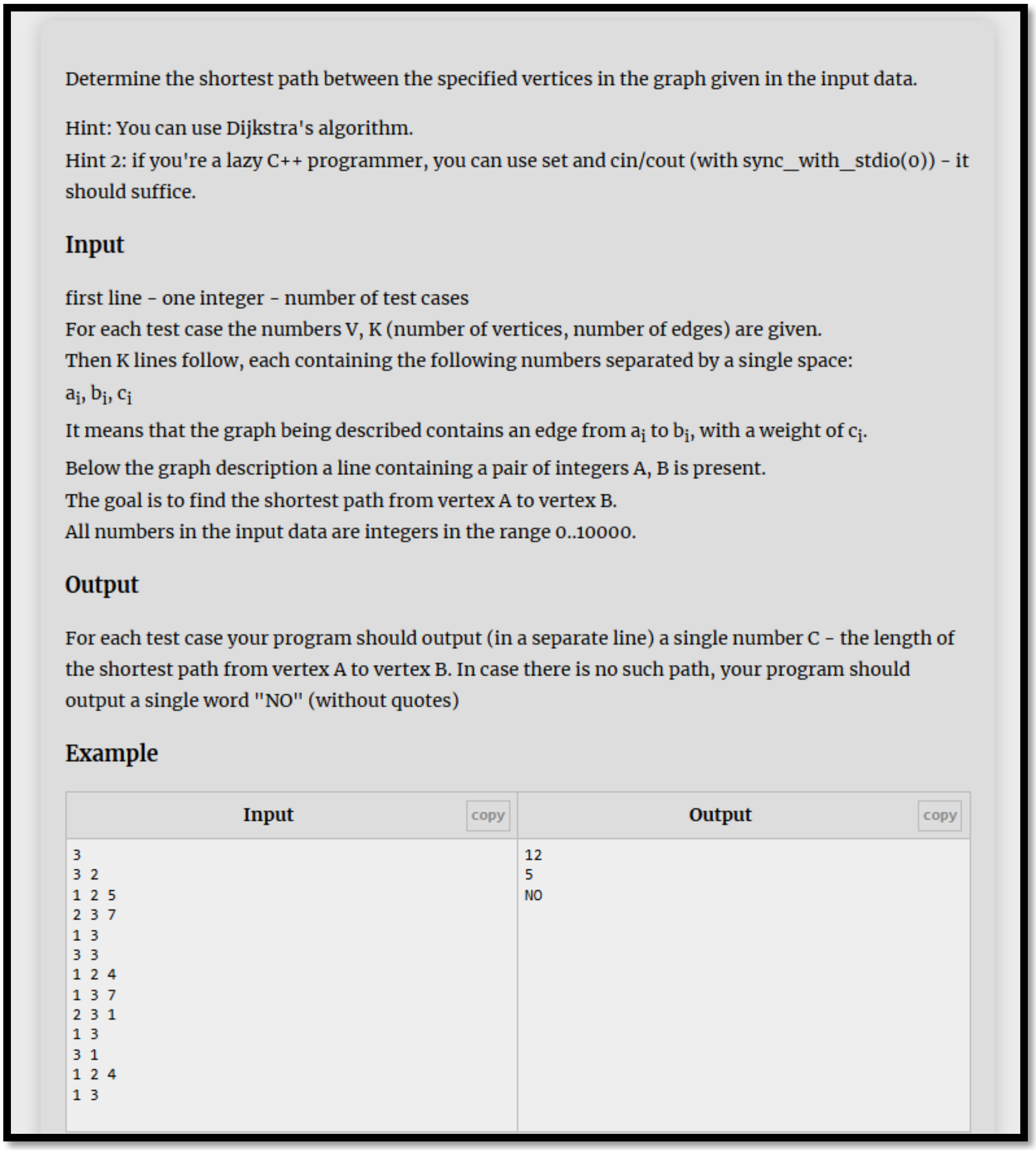
Input copy	Output copy

4 3
2 1
2 3
1 4
	YES

Input copy	Output copy

5 3
3 5
2 4
1 3
	NO

Q2:



Determine the shortest path between the specified vertices in the graph given in the input data.

Hint: You can use Dijkstra's algorithm.
Hint 2: if you're a lazy C++ programmer, you can use set and cin/cout (with sync_with_stdio(0)) - it should suffice.

Input

first line - one integer - number of test cases
For each test case the numbers V, K (number of vertices, number of edges) are given.
Then K lines follow, each containing the following numbers separated by a single space:
a_i, b_i, c_i
It means that the graph being described contains an edge from a_i to b_i, with a weight of c_i.
Below the graph description a line containing a pair of integers A, B is present.
The goal is to find the shortest path from vertex A to vertex B.
All numbers in the input data are integers in the range 0..10000.

Output

For each test case your program should output (in a separate line) a single number C - the length of the shortest path from vertex A to vertex B. In case there is no such path, your program should output a single word "NO" (without quotes)

Example

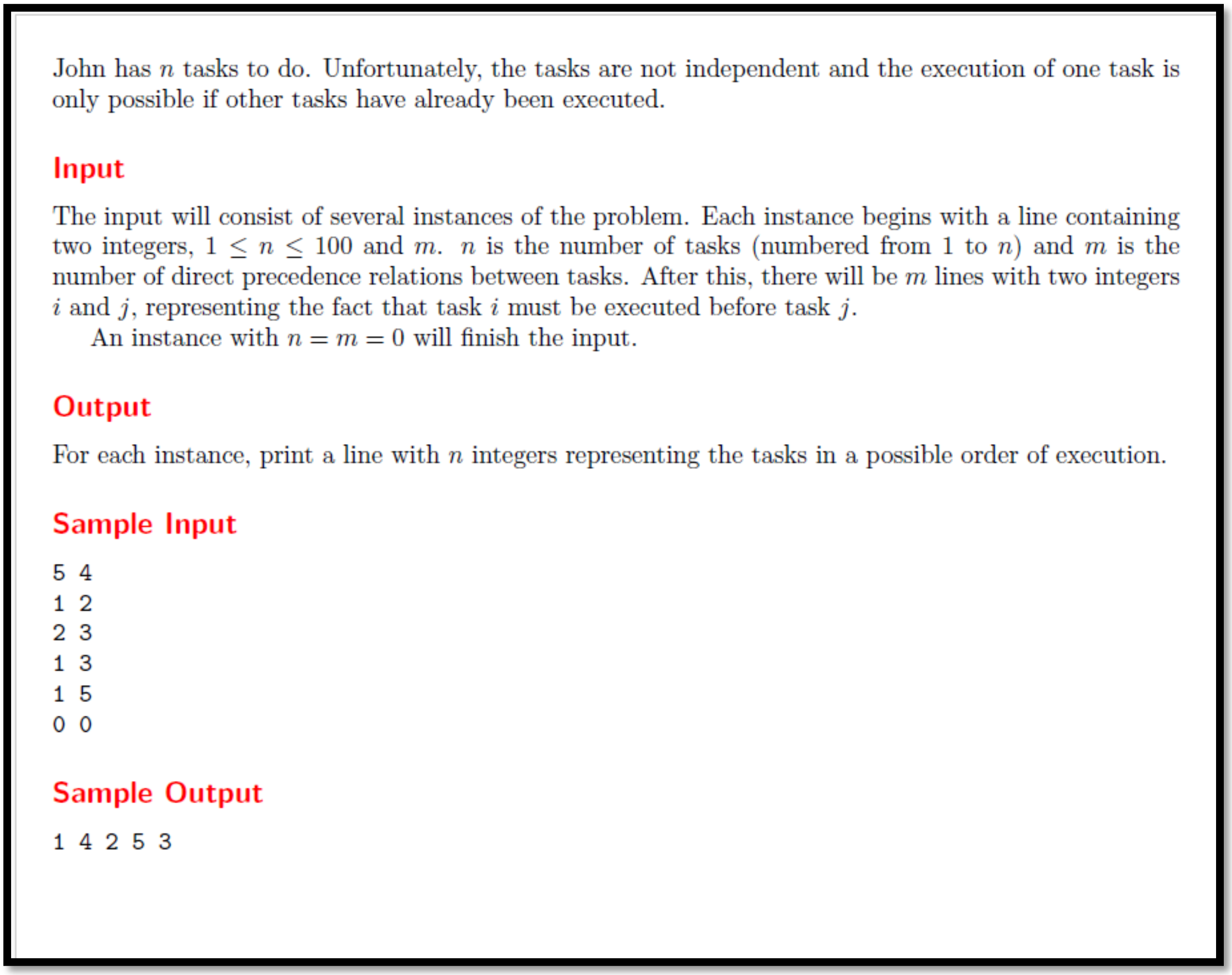
Input

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

Output

```
12
5
NO
```

Q3:



John has n tasks to do. Unfortunately, the tasks are not independent and the execution of one task is only possible if other tasks have already been executed.

Input

The input will consist of several instances of the problem. Each instance begins with a line containing two integers, $1 \leq n \leq 100$ and m. n is the number of tasks (numbered from 1 to n) and m is the number of direct precedence relations between tasks. After this, there will be m lines with two integers i and j, representing the fact that task i must be executed before task j.

An instance with $n = m = 0$ will finish the input.

Output

For each instance, print a line with n integers representing the tasks in a possible order of execution.

Sample Input

```
5 4
1 2
2 3
1 3
1 5
0 0
```

Sample Output

```
1 4 2 5 3
```