

# Kevin and Stones (Easy Version)

Input file:            standard input  
Output file:           standard output  
Time limit:            4 seconds  
Memory limit:         512 megabytes

**This is the easy version of the problem. The difference between the versions is that in this version, you only need to determine whether a valid sequence of operations exists. You can hack only if you solved all versions of this problem.**

Kevin has an undirected graph with  $n$  vertices and  $m$  edges. Initially, some vertices contain stones, which Kevin wants to move to new positions.

Kevin can perform the following operation:

- For each stone at  $u_i$ , select a neighboring vertex  $v_i$ . Simultaneously move each stone from  $u_i$  to its corresponding  $v_i$ .

At any time, each vertex can contain **at most one** stone.

Determine whether a valid sequence of operations exists that moves the stones from the initial state to the target state.

## Input

Each test contains multiple test cases. The first line contains the number of test cases  $t$  ( $1 \leq t \leq 1000$ ). The description of the test cases follows.

The first line of each test case contains two integers  $n$  and  $m$  ( $1 \leq n \leq 2000$ ,  $0 \leq m \leq \min(\frac{n(n-1)}{2}, 10^4)$ ) — the number of vertices and edges in the graph.

The second line contains a binary string  $s$  consisting of '0' and '1'. The  $i$ -th bit of  $s$  indicates the number of stones on the  $i$ -th vertex in the initial state.

The third line contains a binary string  $t$  consisting of '0' and '1'. The  $i$ -th bit of  $t$  indicates the number of stones on the  $i$ -th vertex in the target state.

Each of the next  $m$  lines contains two integers  $u$  and  $v$  ( $1 \leq u, v \leq n$ ) — an undirected edge between the  $u$ -th vertex and the  $v$ -th vertex.

It is guaranteed that the graph is simple. There are no self-loops and parallel edges in the graph.

It is guaranteed that the numbers of '1' in  $s$  and  $t$  are the same.

It is guaranteed that the sum of  $n$  over all test cases does not exceed 2000.

It is guaranteed that the sum of  $m$  over all test cases does not exceed  $10^4$ .

## Output

For each test case, on the first line, output “Yes” or “No” to indicate whether a valid sequence of operations exists.

You can output the answer in any case (upper or lower). For example, the strings “yEs”, “yes”, “Yes”, and “YES” will be recognized as positive responses.

## Example

standard input	standard output
4	Yes
2 1	Yes
10	No
01	Yes
1 2	
11 11	
11011001010	
01101011100	
1 2	
2 3	
3 4	
4 5	
5 6	
6 7	
7 8	
8 9	
9 10	
10 11	
11 1	
3 2	
110	
101	
1 2	
2 3	
3 2	
111	
111	
1 2	
2 3	