

# Kevin and Stones (Hard Version)

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

**This is the hard version of the problem. The difference between the versions is that in this version, you need to output a valid sequence of operations if one 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. Output a valid sequence of operations with no more than  $2n$  moves if one exists. It can be proven that if a valid sequence exists, a valid sequence with no more than  $2n$  moves exists.

## 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.

If a valid sequence of operations exists, output a single integer  $k$  ( $0 \leq k \leq 2n$ ) on the second line, representing the number of operations. Suppose there are  $c$  stones in the initial state. The next  $k + 1$  lines should each contain distinct  $c$  integers, representing the positions of the stones before the operations and after each operation. These positions should satisfy the following:

- The positions of the stones in the first line match the initial state from the input, in any order.

- The positions of the stones in the last line match the target state from the input, in any order.
- For all  $i$  ( $1 \leq i \leq k$ ) and  $j$  ( $1 \leq j \leq c$ ), ensure that the  $j$ -th integer in the  $i$ -th line and the  $j$ -th integer in the  $(i + 1)$ -th line correspond to adjacent vertices in the graph. In other words, the stone is moved from its previous position to the next.

If there are multiple solutions, print any of them.

### Example

standard input	standard output
4	Yes
2 1	1
10	1
01	2
1 2	Yes
11 11	6
11011001010	1 2 4 5 8 10
01101011100	2 3 5 6 9 11
1 2	3 2 6 7 10 1
2 3	4 3 7 8 11 2
3 4	5 2 8 9 1 3
4 5	6 3 7 8 2 4
5 6	7 2 8 9 3 5
6 7	No
7 8	Yes
8 9	0
9 10	1 2 3
10 11	
11 1	
3 2	
110	
101	
1 2	
2 3	
3 2	
111	
111	
1 2	
2 3	