Problem D. Bracket Euler Tour

Input file:	*standard input*
Output file:	*standard output*
Time limit:	2 seconds
Memory limit:	512 mebibytes

You are given an undirected graph with n vertices and m edges. Each vertex has a bracket (either opening "(" or closing ")") associated with it.

You have to find an Euler tour in this graph such that its vertices (written in traversal order) form a correct bracket sequence.

A correct bracket sequence is a sequence of brackets that can be transformed into a correct arithmetic expression by inserting characters "1" and "+" between the original characters of the sequence. For example, bracket sequences "()()", "(())" are correct (the resulting expressions are "(1)+(1)" and "((1+1)+1)"), while ")(" and "()(" are not.

An Euler tour of an undirected graph is a cycle which visits every edge in this graph exactly once. It is allowed to visit the same vertex multiple times, though.

Input

The first line contains two integers n and m ($1 \le n, m \le 2 \cdot 10^5$), the number of vertices and edges respectively.

Each of the following m lines contains two integers v_i and u_i ($1 \le v_i, u_i \le n$), meaning that there is an undirected edge between vertices v_i and u_i. Note that self-loops and multiple edges are allowed.

The last line of the input contains a string of n round brackets, where the i-th bracket is associated with vertex i.

Output

If there is no Euler tour in the given graph that forms a correct bracket sequence, print "No" in the first line.

Otherwise, print "Yes" in the first line. In the second line, print a sequence of vertices that form an Euler tour and also a correct bracket sequence. If there are multiple solutions, print any of them.

Examples

standard input	standard output
2 2 1 2 1 2)(	Yes 2 1
5 6 1 2 2 3 3 1 2 4 4 5 5 2 (()())	Yes 1 2 4 5 2 3
1 1 1 1 (	No