



Voltage 2

Do you know *Just Odd Inventions Co., Ltd.*? This company engages in nothing but just odd inventions. We abbreviate it as JOI.

In one of JOI's laboratories, there is a complicated electric circuit. The circuit consists of N nodes and M thin electric resistors. The nodes are numbered 0 through $N - 1$, and the resistors are numbered 0 through $M - 1$. Each node can be set to one of two states: **high voltage** or **low voltage**. Resistor i ($0 \leq i \leq M - 1$) is connected from node A_i to another node B_i . Current flows through this resistor if and only if node A_i is set to high voltage and node B_i is set to low voltage. In all other cases, no current flows through it. Also, for any two nodes, there is at most one resistor connecting them, regardless of its direction.

You are a researcher at JOI, and you are going to conduct an experiment using this circuit. The resistors in the circuit are so thin that you cannot visually determine which pairs of nodes they connect. However, there is one clue: while the voltages are being set, the temperature of the circuit rises according to the number of resistors through which current flows. Therefore, after setting the voltages, you decide to touch the circuit and read its temperature. Although you cannot measure the exact temperature of the circuit, you can perform two voltage settings and compare which one makes the circuit hotter. That is, by specifying two voltage settings, you can obtain one of the following pieces of information:

- In the first voltage setting, the number of resistors through which current flows is larger.
- In the two voltage settings, the number of resistors through which current flows is the same.
- In the second voltage setting, the number of resistors through which current flows is larger.

Your goal is to repeat such temperature comparisons and determine all resistors in the circuit, that is, all ordered pairs (a, b) such that there is a resistor from node a to node b . You are given in advance the number of nodes N and the number of resistors M . You also know that every resistor connects two distinct nodes, and that for any two nodes there is at most one resistor connecting them, regardless of direction. Under these conditions, determine the entire set of ordered pairs (a, b) corresponding to the resistors in the circuit, from the information obtained by the temperature comparisons. Note that, depending on the structure of the circuit, it may be impossible to determine the resistors uniquely no matter what comparisons are performed or how many are performed. In that case, you must report that it is impossible to determine them uniquely.

To prevent deterioration of the resistors, you are allowed to compare temperatures at most 30 000 times.

Incidentally, what kind of invention JOI is making with this odd circuit is top secret even within the company, and nobody except the president knows it.

Given the numbers of nodes and resistors in the circuit, write a program that determines the resistors in the circuit, or reports that this is impossible, using at most 30 000 temperature comparisons.



Implementation Details

Your submitted program must include `voltage.h` using the `#include` preprocessor directive, and must implement the following function.

- `bool solve(int N, int M)`
 - This function is called exactly once in each execution.
 - The argument `N` represents the number of nodes in the circuit, N .
 - The argument `M` represents the number of resistors in the circuit, M .
 - This function must return `false` if the resistor connections cannot be uniquely determined no matter what temperature comparisons are performed or how many are performed; otherwise, it must return `true`.
 - If this function returns `true` when the resistor connections cannot be determined uniquely, your program is judged as **Wrong Answer [1]**.
 - If this function returns `false` even though the circuit can be determined from temperature comparisons, your program is judged as **Wrong Answer [2]**.

Your program may call the following functions.

- ★ `int query(std::vector<int> x, std::vector<int> y)`
 - By using this function, you can perform two voltage settings and compare their temperatures.
 - The argument `x` specifies the first voltage setting, and the argument `y` specifies the second voltage setting.
 - The arguments `x` and `y` must be arrays of length N consisting of 0's and 1's.
 - If `x[k]` ($0 \leq k \leq N - 1$) is 1, then in the first voltage setting, node k is set to high voltage; if `x[k]` is 0, then node k is set to low voltage.
 - If `y[k]` ($0 \leq k \leq N - 1$) is 1, then in the second voltage setting, node k is set to high voltage; if `y[k]` is 0, then node k is set to low voltage.
 - The return value of this function is the result of comparing the temperatures of the circuit under the first and second voltage settings, and is one of -1 , 0 , or 1 .
 - ◊ If the return value is -1 , then the number of resistors through which current flows is larger in the first voltage setting than in the second.
 - ◊ If the return value is 0 , then the numbers of resistors through which current flows are equal in the first and second voltage settings.
 - ◊ If the return value is 1 , then the number of resistors through which current flows is larger in the



second voltage setting than in the first.

- If the length of x is not N , your program is judged as **Wrong Answer [3]**.
- If x contains a value other than 0 or 1, your program is judged as **Wrong Answer [4]**.
- If the length of y is not N , your program is judged as **Wrong Answer [5]**.
- If y contains a value other than 0 or 1, your program is judged as **Wrong Answer [6]**.
- You must not call this function more than 30 000 times. If this function is called more than 30 000 times, your program is judged as **Wrong Answer [7]**.

★ `void answer(int a, int b)`

- Use this function to report a resistor that you have identified.
- The arguments a and b indicate that there is a resistor directed from node a to node b .
- It must hold that $0 \leq a \leq N - 1$ and $0 \leq b \leq N - 1$. If this condition is not satisfied, your program is judged as **Wrong Answer [8]**.
- You must not call this function two or more times with the same pair (a, b) . If this condition is violated, your program is judged as **Wrong Answer [9]**.
- You must not call this function more than M times. If this function is called more than M times, your program is judged as **Wrong Answer [10]**.
- When the function `solve` returns `true`, the function `answer` must have been called exactly M times by then. If this condition is not satisfied, your program is judged as **Wrong Answer [11]**.
- When the function `solve` returns `true`, for every pair (a, b) passed as arguments to `answer` up to that point, there must actually exist a resistor directed from node a to node b . If this condition is not satisfied, your program is judged as **Wrong Answer [12]**.

Important Notices

- You may implement additional helper functions or declare global variables for internal use.
- Your submitted program must not use standard input/output or any other file interaction. However, you may use standard error output for debugging.



Compilation and Execution

You can download an archive file from the contest webpage which contains the sample grader to test your program. The archive file also contains a sample source file of your program.

The sample grader is the file `grader.cpp`. To test your program, place the files `grader.cpp`, `voltage.cpp`, and `voltage.h` in the same directory, and compile them using the following command:

```
g++ -std=gnu++20 -O2 -o grader grader.cpp voltage.cpp
```

Alternatively, you can execute the `compile.sh` file included in the archive by running:

```
./compile.sh
```

If compilation is successful, an executable file named `grader` will be generated.

Note that the actual grader is different from the sample grader. The sample grader is launched as a single process, which will read input data from the standard input and write the results to the standard output.

Input for the Sample Grader

The sample grader reads input from standard input in the following format:

```
 $N$   $M$   
 $A_0$   $B_0$   
⋮  
 $A_{M-1}$   $B_{M-1}$ 
```

Output of the Sample Grader

The sample grader outputs the following information to standard output (quotation marks are not included in the actual output).

- If your program is judged as one of Wrong Answer [3] – [12], the type of Wrong Answer is printed as in “Wrong Answer [5]”.
- Otherwise, the number of calls to the function `query` and the return value of the function `solve` are printed as in “Accepted: 30 true”. The sample grader, unlike the actual grader, does not judge whether the verdict should be Wrong Answer [1] or [2], that is, whether the return value of the function



solve is correct.

The sample grader terminates execution as soon as any of the conditions for Wrong Answer [3] – [12] is met. If multiple incorrect conditions are met, only one of them is displayed.

Notices on Grading

The actual grader is not adaptive; it has a fixed answer from the beginning of the interaction.

Constraints

- $2 \leq N \leq 500$.
- $1 \leq M \leq 1\,000$.
- $0 \leq A_i \leq N - 1$ ($0 \leq i \leq M - 1$).
- $0 \leq B_i \leq N - 1$ ($0 \leq i \leq M - 1$).
- $A_i \neq B_i$ ($0 \leq i \leq M - 1$).
- $(A_i, B_i) \neq (A_j, B_j)$ and $(A_i, B_i) \neq (B_j, A_j)$ ($0 \leq i < j \leq M - 1$).
- N, M, A_i , and B_i are integers ($0 \leq i \leq M - 1$).

Subtasks

1. (10 points) $N \leq 100$, $M = N - 1$, $B_i = A_{i+1}$ ($0 \leq i \leq N - 3$), and the N values $A_0, A_1, \dots, A_{N-2}, B_{N-2}$ are all distinct.
2. (12 points) $M = N - 1$, $B_i = A_{i+1}$ ($0 \leq i \leq N - 3$), and the N values $A_0, A_1, \dots, A_{N-2}, B_{N-2}$ are all distinct.
3. (27 points) $N \leq 100$, $A_i \neq A_j$ ($0 \leq i < j \leq M - 1$).
4. (18 points) $A_i \neq A_j$ ($0 \leq i < j \leq M - 1$).
5. (17 points) $N \leq 100$.
6. (16 points) No additional constraints.



Example Interaction

Below is an example of input that the sample grader reads and the corresponding function calls.

Sample Input 1	solve Call	Return Value	Function Call	Return Value
	solve(5, 6)			
5 6			query([0, 0, 1, 1, 1], [1, 1, 1, 0, 0])	-1
0 2			query([1, 0, 1, 0, 0], [0, 1, 0, 1, 0])	0
2 1			query([0, 1, 1, 1, 0], [1, 1, 0, 1, 1])	1
0 3			answer(0, 2)	
3 2			answer(0, 3)	
3 4			answer(2, 1)	
4 1			answer(3, 4)	
			answer(3, 2)	
			answer(4, 1)	
		true		

In the first call to query, the two voltage settings and the resistors through which current flows are as follows.

- In the first setting, nodes 0 and 1 are set to low voltage, and nodes 2, 3, and 4 are set to high voltage. At this time, current flows through resistors 1 and 5.
- In the second setting, nodes 3 and 4 are set to low voltage, and nodes 0, 1, and 2 are set to high voltage. At this time, current flows through resistor 2.

Since the number of resistors through which current flows is larger in the first voltage setting, the return value is -1 .

This sample input satisfies the constraints for subtasks 5, 6.

Among the files available for download from the contest site: `sample-01-in.txt` corresponds to Sample Input 1. Additionally, `sample-02-in.txt` satisfies the constraints for all subtasks, and `sample-03-in.txt` satisfies the constraints for subtasks 3, 4, 5, 6.