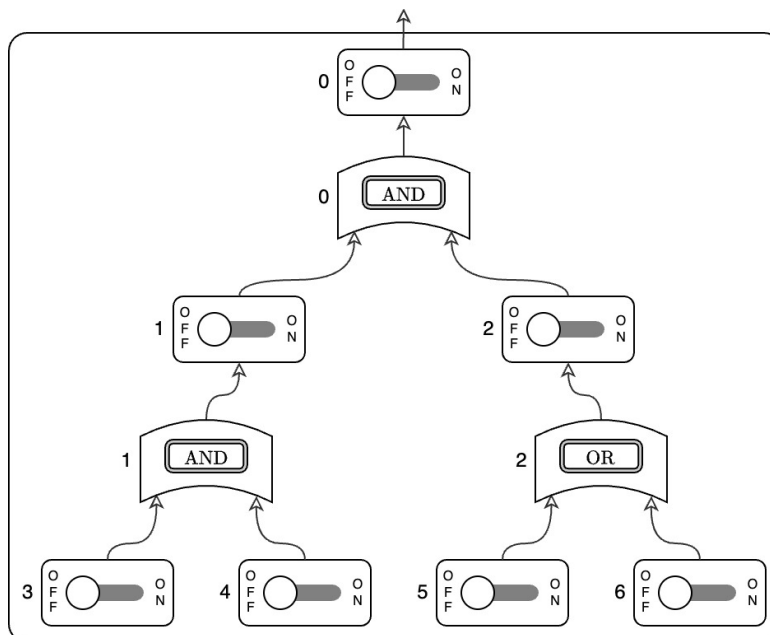# Circuit 2

JOI-kun is playing with an electronic circuit set.

The electronic circuit set consists of N **AND components**, N **OR components**, and one **circuit board**. The circuit board is composed of $2N + 1$ **switches** and N **component slots**, where each component slot can be used by placing either an AND component or an OR component. The circuit board outputs a value of 0 or 1, depending on the placed components and the states of the switches.

Specifications of the Circuit Board

- Each switch is assigned a number from 0 to $2N$, and each switch has an ON or OFF state. Each switch outputs a value of 0 or 1 as described below.
- Each component slot is assigned a number from 0 to $N - 1$. Each component slot also outputs a value of 0 or 1 as described below.
- The output of each switch and each component slot is determined in order from the highest-numbered one to the lowest, according to the following rules. If a switch and a component slot share the same number, the component slot's output is determined first.
    - For $j = 2N, 2N - 1, \ldots, N$, the output of switch j is determined as follows:
        - If switch j is OFF, it outputs 0.
        - If switch j is ON, it outputs 1.
    - For $j = N - 1, N - 2, \ldots, 0$, let the output of component slot j be x. The output of switch j is determined as follows:
        - If switch j is OFF, it outputs x.
        - If switch j is ON, it outputs $1 - x$.
    - For $i = N - 1, N - 2, \ldots, 0$, component slot i is connected to two switches, U_i and V_i, where $i < U_i < V_i \le 2N$. Let the output of switch U_i be x and the output of switch V_i be y. The output of component slot i is determined as follows:
        - If component slot i has an AND component, it outputs $\min(x, y)$.
        - If component slot i has an OR component, it outputs $\max(x, y)$.
- For each $j = 1, 2, \ldots, 2N$, there exists exactly one i ($0 \le i \le N - 1$) such that $U_i = j$ or $V_i = j$.
- The output of the circuit board is equal to the output of switch 0.

For example, when $N = 3$, $U_0 = 1$, $V_0 = 2$, $U_1 = 3$, $V_1 = 4$, $U_2 = 5$, $V_2 = 6$, and AND components are placed in component slots 0 and 1 while an OR component is placed in component slot 2, the circuit board is represented as shown in the figure below.

Now, JOI-kun attempted to place AND components in all component slots. However, it turned out that **up to R OR components** were accidentally mixed in. Since AND and OR components look identical, they must be distinguished using the circuit board. Your task is to identify which component slots contain OR components by asking at most $1\,000$ queries in the following format:

- You can instruct JOI-kun on how to set the $2N + 1$ switches. JOI-kun will then configure the switches accordingly and report the circuit board's output to you.

Given the connection structure of the circuit board and the upper bound on the number of OR components, write a program that determines the locations of all OR components using at most $1\,000$ queries.

Implementation Details

You need to submit one file.

The file is `circuit.cpp`. This program should include `circuit.h` using the `#include` preprocessor directive.

`circuit.cpp` must implement the following function:

- `std::string solve(int N, int R, std::vector<int> U, std::vector<int> V)`
    - This function is called exactly once per test case.
    - The argument `N` represents the number of component slots, N.
    - The argument `R` is the upper bound on the number of OR components, R.
    - The arguments `U` and `V` are arrays of length N, where `U[i]` and `V[i]` ($0 \le i \le N - 1$) are the switch numbers U_i, V_i connected to component slot i.
    - This function must return a string t of length N consisting of '`&`' and '`|`' that satisfies the following condition:
        - For each $i = 0, 1, \ldots, N - 1$, if component slot i contains an AND component, then `t[i]` must be '`&`'. If component slot i contains an OR component, then `t[i]` must be '`|`'.
    - If the returned string t is not of length N, your program is judged as **Wrong Answer [1]**.
    - If t contains characters other than '`&`' or '`|`', your program is judged as **Wrong Answer [2]**.
    - If the actual type of component placed in each slot does not match the one represented by the returned string t, your program is judged as **Wrong Answer [3]**.

Your program may call the following function:

- ★ `int query(std::string s)`
    - You can use this function to ask JOI-kun a query.
    - The argument `s` must be a string of length $2N + 1$ consisting only of '`0`' and '`1`'. For each $j = 0, 1, \ldots, 2N$, if `s[j]` is '`0`', switch j is set to OFF. If `s[j]` is '`1`', switch j is set to ON.
    - If the length of `s` is not $2N + 1$, your program is judged as **Wrong Answer [4]**.
    - If `s` contains characters other than '`0`' or '`1`', your program is judged as **Wrong Answer [5]**.
    - You must not call this function more than 1 000 times. If this function is called more than 1 000 times, your program is judged as **Wrong Answer [6]**.
    - The return value of this function is the output of the circuit board after setting the switches as specified in `s`.

The 24th Japanese Olympiad in Informatics (JOI 2024/2025)
Spring Training/Qualifying Trial
March 20–24, 2025 (Komaba, Tokyo)

Contest 4 – Circuit 2

Important Notices

- You may implement additional helper functions or declare global variables for internal use.
- Your 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`, `circuit.cpp`, and `circuit.h` in the same directory, and compile them using the following command:

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

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

```
./compile.sh
```

If the 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 will be executed 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

Let T be the string of length N that the function `solve` should return. The sample grader reads input from standard input in the following format:

$N\ R$
$U_0\ V_0$
$U_1\ V_1$
$\vdots$
$U_{N-1}\ V_{N-1}$
T

The 24th Japanese Olympiad in Informatics (JOI 2024/2025)
Spring Training/Qualifying Trial
March 20–24, 2025 (Komaba, Tokyo)

Contest 4 – Circuit 2

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 correct, it reports the number of calls to `query` as "`Accepted: 22`".
- If your program is judged as any type of Wrong Answer, the sample grader writes its type as "`Wrong Answer [4]`".

The sample grader terminates execution as soon as any incorrect condition 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

- $1 \leq N \leq 8\,000$.
- $1 \leq R \leq \min(N, 120)$.
- $i < U_i < V_i \leq 2N$ ($0 \leq i \leq N - 1$).
- For each $j = 1, 2, \ldots, 2N$, there exists exactly one i ($0 \leq i \leq N - 1$) such that $U_i = j$ or $V_i = j$.

Subtasks

1. (1 point) $N = 1$.
2. (4 points) $N \leq 1\,000$, $R = 1$.
3. (5 points) $N \leq 1\,000$.
4. (17 points) $U_i = i + 1$, $V_i = N + 1 + i$ ($0 \leq i \leq N - 1$), $R \leq 70$.
5. (8 points) $U_i = i + 1$, $V_i = N + 1 + i$ ($0 \leq i \leq N - 1$).
6. (23 points) $U_i = 2i + 1$, $V_i = 2i + 2$ ($0 \leq i \leq N - 1$), $R \leq 70$.
7. (8 points) $U_i = 2i + 1$, $V_i = 2i + 2$ ($0 \leq i \leq N - 1$).
8. (27 points) $R \leq 70$.

9. (7 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
1 1
1 2
|

solve Call	Return Value	query Call	Return Value
solve(1, 1, [1], [2])			
		query("010")	1
		query("011")	1
		query("111")	0
	"|"		

In the first call to query, the output of the circuit board can be calculated as follows:

- Switches 1 and 2 are ON and OFF, respectively, so switch 1 outputs 1 and switch 2 outputs 0.
- Component slot 0 contains an OR component, and the connected switches 1 and 2 output 1 and 0, respectively. Thus, component slot 0 outputs $\max(1, 0) = 1$.
- Switch 0 is OFF, and component slot 0 outputs 1, so switch 0 outputs 1.
- Therefore, the circuit board outputs 1.

This example satisfies the constraints of all subtasks.

Sample Input 2
3 3
1 2
3 4
5 6
&&|

solve Call	Return Value	query Call	Return Value
solve(3, 3, [1, 3, 5], [2, 4, 6])			
		query("0001001")	0
		query("0001110")	1
		query("0000011")	0
	"&&|"		

The diagram in the problem statement represents the circuit board for this example.

This example satisfies the constraints of subtasks $3, 6, 7, 8, 9$.

Among the files available for download from the contest site: sample-01-in.txt corresponds to Sample

Input 1, and `sample-02-in.txt` corresponds to Sample Input 2. Additionally, `sample-03-in.txt` satisfies the constraints of subtasks $3, 4, 5, 8, 9$, and `sample-04-in.txt` satisfies the constraints of subtasks $3, 6, 7, 8, 9$.