



WELCOME TO  
**COMPETITIVE PROGRAMMING CELL**

BY : WIAME JAQUI & ALAE CAICH

# COMPETITIVE PROGRAMMING



Trying to do better than others — it's about speed, accuracy, and problem-solving.

Telling a computer what to do by writing instructions in a language it understands.

# PROBLEM STRUCTURE

## STATEMENT & CONSTRAINTS

### H. Beautiful Problem

time limit per test: 2 seconds

memory limit per test: 256 megabytes

For an array  $a$  of length  $n$  and three integers  $x$ ,  $l$ , and  $r$  ( $1 \leq l \leq r \leq n$ ), define:

$$f(a, x, l, r) = \begin{cases} 0, & \text{if } (x - \min_{j=l}^r(a_j)) \cdot (x - \max_{j=l}^r(a_j)) < 0 \\ 1, & \text{if } (x - \min_{j=l}^r(a_j)) \cdot (x - \max_{j=l}^r(a_j)) \geq 0 \end{cases}$$

You are given an array  $a$  of length  $n$  ( $1 \leq a_i \leq n$ ), and  $m$  intervals  $[l_i, r_i]$  ( $1 \leq l_i \leq r_i \leq n$ ).

For each  $x = 1, 2, \dots, n$ , answer the following question independently:

- Does there exist a rearrangement  $a'$  of  $a$ , such that for all  $1 \leq i \leq m$ ,  $f(a', x, l_i, r_i) = 1$ ?

## INPUT

### Input

The first line contains a single integer  $t$  ( $1 \leq t \leq 2 \cdot 10^4$ ) — the number of test cases. Description of each testcase follows.

The first line contains two integers  $n$  and  $m$  ( $2 \leq n \leq 2000$ ,  $1 \leq m \leq 2000$ ).

The next line contains  $n$  space-separated integers  $a_1, a_2, \dots, a_n$  ( $1 \leq a_i \leq n$ ).

The next  $m$  lines each contain two space-separated integers  $l_i, r_i$  ( $1 \leq l_i \leq r_i \leq n$ ), each denoting an interval.

It is guaranteed that the sum of  $n^2$  and the sum of  $m^2$  over all test cases does not exceed  $4 \cdot 10^6$ , respectively.

## OUTPUT

### Output

For each test case, output a binary string  $s$ . For  $x = 1, 2, \dots, n$ ,  $s_x = 1$  only if there exists a rearrangement  $a'$  of  $a$ , such that for all  $1 \leq i \leq m$ ,  $f(a', x, l_i, r_i) = 1$ . Otherwise,  $s_x = 0$ .

## EXAMPLES

### Example

#### input

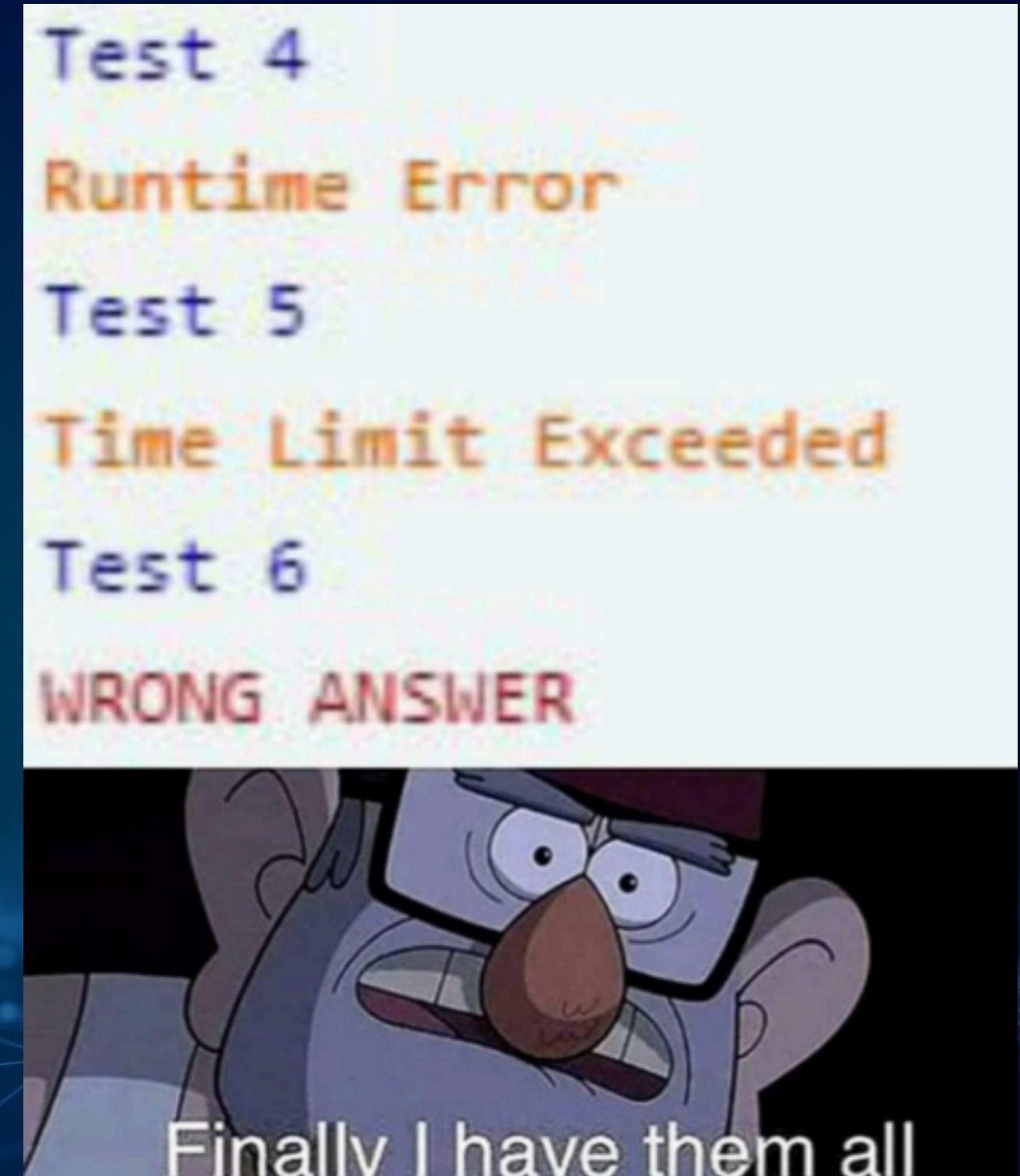
```
4
4 2
1 1 3 4
1 2
2 4
3 2
1 1 3
1 2
2 3
3 1
1 1 1
1 3
9 3
4 5 9 1 1 1 2 2 3
1 6
3 7
7 9
```

#### output

```
1011
101
111
100100001
```

# ALGORITHM

When solving a problem, you need to make sure your solution works for all possible test cases, not just one — and it must do so within the time and memory limits!



# WANT TO BE A CP PROGRAMMER ?

**01**

Basic requirements

- A basic understanding of at least 1 coding language (preferably C++)
- Being ready to practice, ehm.. a lot

**02**

Favorable requirements

- A good mathematical background
- A good analytical sense
- Problem-solving skills
- A good knowledge of data structures
- Teamwork skills

# WHERE TO TRAIN ?

 POJ	 ZOJ	 UVa UVALive	 SGU
 URAL	 HUST	 SPOJ	 HDU
 HYSBZ	 UVA	 CodeForces	 Z-Trening
 Aizu	 LightOJ	 UESTC	 NBUT
 FZU	 CSU	 SCU	 ACdream
 CodeChef	 CF::Gym	 OpenJudge	 Kattis
 HihoCoder	 HIT	 HRBUST	 EIJudge
 AtCoder	 HackerRank	 51Nod	 TopCoder
 E-Olymp	 计蒜客	 LibreOJ	 UniversalOJ
 黑暗爆炸	 CSG	 DMOJ	 Toph
 洛谷	 Baekjoon	 QOJ	 CSES

CODE

# C++ LANGUAGE

## WHAT IS C++ ?

- Game development (Unreal Engine)
- Operating systems
- Embedded systems
- Competitive programming



CODE

Oct 2025	Oct 2024	Change	Programming Language	Ratings	Change
1	1		 Python	24.45%	+2.55%
2	4		 C	9.29%	+0.91%
3	2		 C++	8.84%	-2.77%
4	3		 Java	8.35%	-2.15%
5	5		 C#	6.94%	+1.32%
6	6		 JavaScript	3.41%	-0.13%
7	7		 Visual Basic	3.22%	+0.87%
8	8		 Go	1.92%	-0.10%
9	10		 Delphi/Object Pascal	1.86%	+0.19%
10	11		 SQL	1.77%	+0.13%
11	9		 Fortran	1.70%	-0.10%
12	29		 Perl	1.66%	+1.10%
13	17		 R	1.52%	+0.43%
14	15		 PHP	1.38%	+0.17%

CODE

# C++ LANGUAGE

## WHY C++ ?

- Speed — C++ is faster than most languages.
- Powerful Standard Library (STL) — ready-to-use algorithms & data structures.
- Used in most contests (Codeforces, AtCoder, ICPC...).
- Supports object-oriented programming



# BASIC SYNTAX

```
#include <iostream>
using namespace std;

int main() {
    cout << "Hello, World!" << endl;
    return 0;
}
```

# VARIABLES & DATA TYPES

```
int age = 18;  
float height = 1.75;  
char grade = 'A';  
string name = "Alae";  
bool isStudent = true;
```

# CONDITIONAL(IF-ELSE)

```
int grade = 75;

if (grade >= 90)
    cout << "A";
else if (grade >= 80)
cout << "B";
    else if (grade >= 70)
cout << "C";
else
    cout << "F";
```

# LOOPS(FOR LOOP)

```
for (int i = 1; i <= 5; i++) {  
    cout << i << " ";  
}
```

# LOOPS(WHILE LOOP)

```
while (i <= 5) {  
    cout << i << " ";  
    i++;  
}
```

# LOOPS(DO-WHILE LOOP)

```
int i = 1;  
do {  
    cout << i << " "  
    i++;  
} while (i <= 5);
```

# BREAK AND CONTINUE STATEMENTS

```
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        break;  
    }  
    cout << i << " ";  
}
```

```
for (int i = 1; i <= 5; i++) {  
    if (i == 3) {  
        continue;  
    }  
    cout << i << " ";  
}
```

# CONDITIONAL(SWITCH)

```
int day = 3;

switch(day) {
    case 1: cout << "Monday";
    break;
    case 2: cout << "Tuesday";
    break;
    case 3: cout << "Wednesday";
    break;
    default: cout << "Invalid day";
}
```

# FUNCTIONS

```
int add(int a, int b) {  
    return a + b;  
}
```

# CONCLUSION