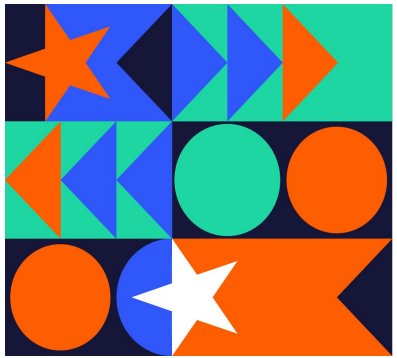
INTRODUCCIÓN A LA PROGRAMACIÓN COMPETITIVA CON KOTLIN



KOTLIN
HEROES

COMPETITIVE
PROGRAMMING

PRESENTACIÓN

Jorge Rodríguez Castillo

Software Developer, enfocado en desarrollo de aplicaciones móviles para Android

Estudios en Inteligencia Artificial (PUCP), enfocado en Deep Learning.

Hobbies: Natación, ver series, programación competitiva.

jjorge.rc93@gmail.com

https://www.linkedin.com/in/jorge-rodríguez-castillo/

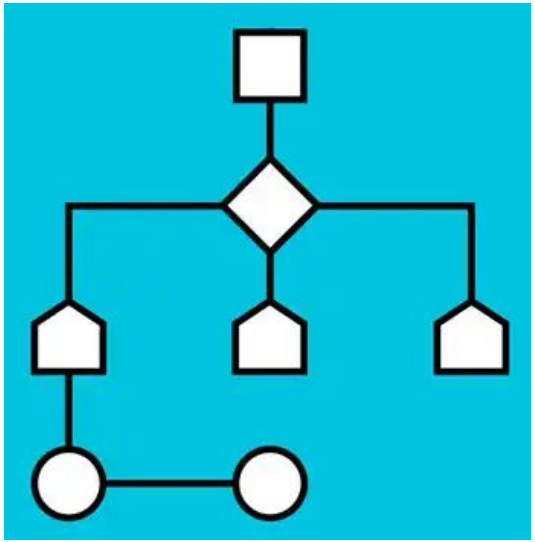
https://github.com/jjrodcast

¿QUÉ ES PROGRAMACIÓN COMPETITIVA?

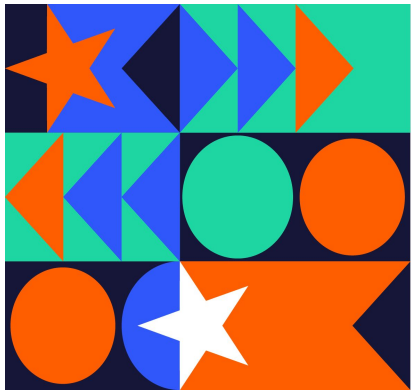
Es un deporte mental practicado de manera general en plataformas de internet y en algunas ocasiones de forma presencial.

La idea principal es tratar de resolver problemas abstraídos de la realidad en soluciones rápidas y eficientes usando distintas estructuras de datos en nuestros algoritmos.

Pensar como si fuera un puzzle a resolver ayuda un poco.



KOTLIN HEROES



KOTLIN
HEROES

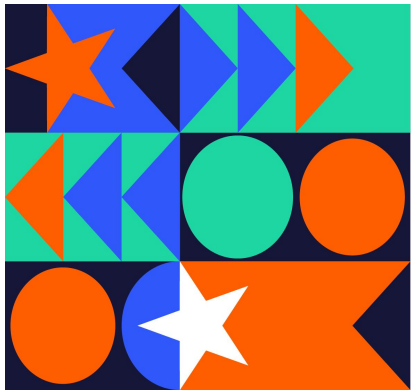
Kotlin Heroes es una iniciativa de JetBrains y usa **Kotlin** como lenguaje para resolver problemas de algoritmia.

La idea es **retar tus conocimientos** sobre una serie de tareas y además llevar tus conocimientos de programación a un nivel superior. **No es necesario tener conocimientos avanzados.**

Kotlin Heroes ya tiene 4 Episodios los cuales están alojados en Codeforces

(*) No podemos usar ninguna librería externa a Kotlin

KOTLIN HEROES 4: Practice & Contest



KOTLIN
HEROES

Practice

Problems

#	Name
A	A+B (Trial Problem)
B	Square?
C	Sum of Round Numbers
D	Alice, Bob and Candies
E	Special Permutation

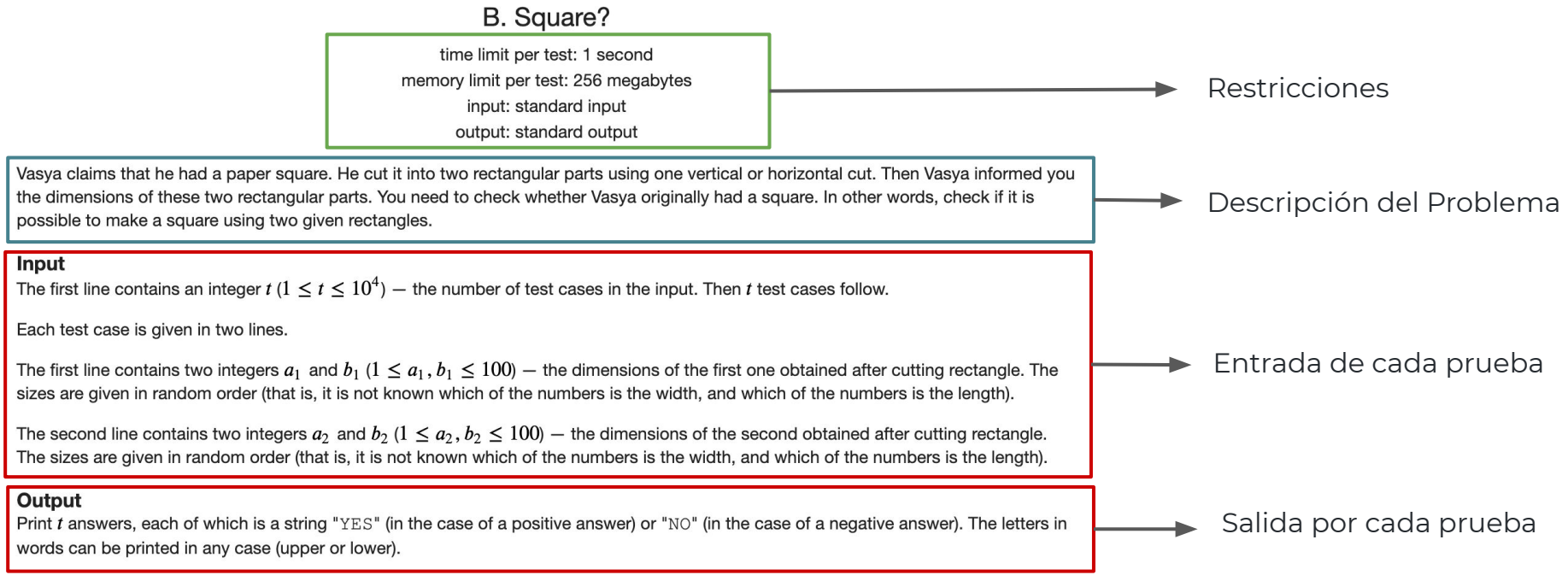
Contest

Problems

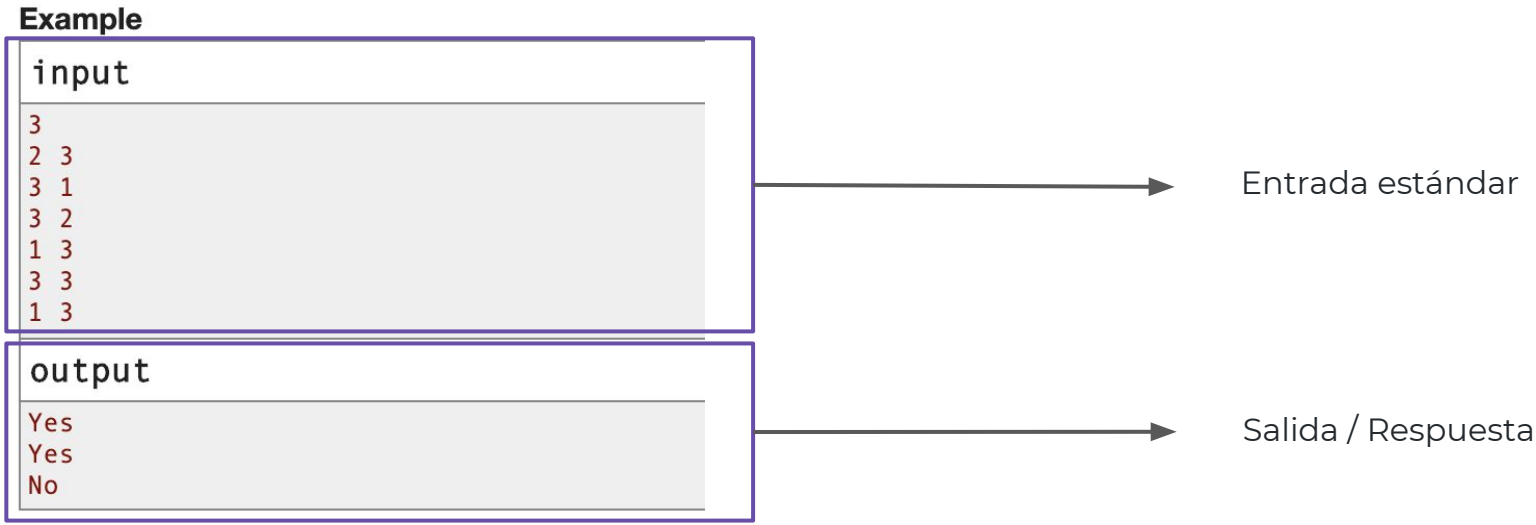
#	Name
A	Color Revolution
B	Boot Camp
C	Spring Cleaning
D	Constructing the Dungeon
E	Magic Tricks
F	Dune II: Battle For Arrakis
G	Two IP Cameras
H	Game with Segments
I	Pac-Man 2.0

(*) No podemos usar ninguna librería externa a Kotlin

ESTRUCTURA DE UN PROBLEMA



ESTRUCTURA DE UN PROBLEMA



ANALIZAMOS EL PROBLEMA

Vasya claims that he had a paper square. He cut it into two rectangular parts using one vertical or horizontal cut. Then Vasya informed you the dimensions of these two rectangular parts. You need to check whether Vasya originally had a square. In other words, check if it is possible to make a square using two given rectangles.

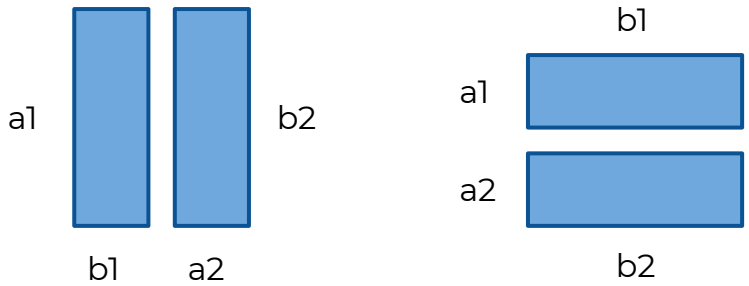
Input

The first line contains an integer t ($1 \le t \le 10^4$) — the number of test cases in the input. Then t test cases follow.

Each test case is given in two lines.

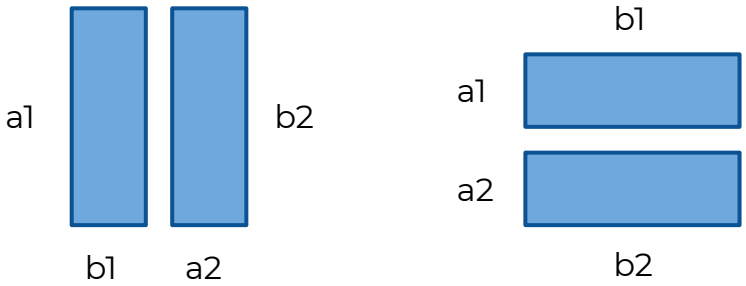
The first line contains two integers a_1 and b_1 ($1 \le a_1, b_1 \le 100$) — the dimensions of the first one obtained after cutting rectangle. The sizes are given in random order (that is, it is not known which of the numbers is the width, and which of the numbers is the length).

The second line contains two integers a_2 and b_2 ($1 \le a_2, b_2 \le 100$) — the dimensions of the second obtained after cutting rectangle. The sizes are given in random order (that is, it is not known which of the numbers is the width, and which of the numbers is the length).



RESOLVEMOS EL PROBLEMA (localmente)

```kotlin
package pe.com.jjrodcast.competitiveprogramming

private fun readLn() : String  = readLine()!!
private fun readInt() : Int  = readLn().toInt()
private fun readStrings() : List<String>  = readLn().split( ...delimiters: " ")
private fun readInts() : List<Int>  = readStrings().map { it.toInt() }

private fun wasSquare(a1: Int, b1: Int, a2: Int, b2: Int): Boolean {
    /*
        Condición 1: El valor de  `width` o `height` de R1 de ser igual al de R2

        Condición 2: Suma del mínimo valor de `width`/`height` R1 y R2,
                     debe ser igual al maximo valor (`width`,`height`) de R2
    */
    val condition1 : Boolean  = maxOf(a1, b1) == maxOf(a2, b2)
    val condition2 : Boolean  = minOf(a1, b1) + minOf(a2, b2) == maxOf(a1, b1)

    return condition1 && condition2
}

fun main() {
    val t : Int  = readInt()
    for (i : Int  in 0 until t) {
        val (a1 : Int , b1 : Int ) = readInts()
        val (a2 : Int , b2 : Int ) = readInts()
        if (wasSquare(a1, b1, a2, b2)) println("Yes") else println("No")
    }
}
```



Example

input
3 2 3 3 1 3 2 1 3 3 3 1 3

output
Yes Yes No

SUBIMOS NUESTRA SOLUCIÓN

Vamos a Codeforces

RESULTADOS QUE PODEMOS OBTENER

Judgement Verdict	Description
Memory limit exceeded	The program tries to consume more memory than is indicated in the problem statement
Time limit exceeded	The program hadn't terminated in time indicated in the problem statement
Runtime error	The program terminated with a non-zero return code (possible reasons: array out of bound error, division by zero, stack overflow, incorrect pointers usage, etc)
Wrong answer	Wrong answer

TIPOS DE ALGORITMOS

- Ad-hoc
- Teoría de números
- Combinatorias
- Estructuras de datos
- Grafos
- Teoría de Juegos
- Programación Dinámica
- Geometría Computacional

OPCIONES PARA PRACTICAR

URI ONLINE JUDGE
PROBLEMS & CONTESTS

https://www.urionlinejudge.com.br

LeetCode

https://leetcode.com/

HackerRank

https://www.hackerrank.com/

codewars

https://www.codewars.com/

EJEMPLO EN LEETCODE

63. Unique Paths II

Medium 2033 254 Add to List Share

A robot is located at the top-left corner of a *m* x *n* grid (marked 'Start' in the diagram below).

The robot can only move either down or right at any point in time. The robot is trying to reach the bottom-right corner of the grid (marked 'Finish' in the diagram below).

Now consider if some obstacles are added to the grids. How many unique paths would there be?

An obstacle and empty space is marked as `1` and `0` respectively in the grid.

Note: *m* and *n* will be at most 100.

Example 1:

```
Input:
[
  [0,0,0],
  [0,1,0],
  [0,0,0]
]
Output: 2
```

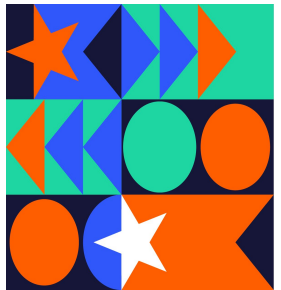
LeetCode

LEETCODE

Hint: Pre calcular los 'paths' en un arreglo bidimensional

RECOMENDACIONES

- Tener ganas de realizar y practicar algoritmos; ya sean fáciles o difíciles.
- Si no tienes experiencia no te sientas mal, todos empezamos sin tener muchos conocimientos.
- No lo tomes como algo tedioso, sino como una forma de entrenar y desarrollar tus habilidades de programación.



KOTLIN
HEROES

GRACIAS