

#CátedrasCiber

Módulo I: Introducción

19/02/2025



Información del curso

1. Horario de clase.
2. Material básico necesario.
3. Módulos del curso.
4. Plataformas que vamos a utilizar.

Horario y enlaces

- 19 Febrero – 26 Marzo (5 sesiones).
- Los miércoles de 17:00 a 19:00 (excepto el último).
- Online: Se anunciará el enlace de la sesión por correo cada semana y si no la web estará actualizada.
- Concurso final presencial.
- Se grabarán las sesiones para aquellos que no puedan venir.
- Información: <https://catedra-uc-incibe.github.io/curso-cp/>

¿Qué es necesario para participar?

- Un entorno de desarrollo con Python instalado (podéis usar otros lenguajes, pero el curso se enfoca en Python).
- Ganas de aprender.
- ¿Conocimientos básicos en algún lenguaje de programación?
- ¿Experiencia en entrevistas técnicas? ¿Conocimientos de jueces online de código?

Módulos del curso

- Módulo I: Introducción.
- Módulo II: Estructuras de datos y fallos comunes.
- Módulo III: Grafos en ciberseguridad.
- Módulo IV: Fuerza bruta y programación dinámica.

Las sesiones serán teórico-prácticas con retos durante la sesión y más retos después de la sesión, el propósito es aprender a usar las herramientas y aplicar el conocimiento.

Plataforma de problemas del curso

<https://cp-catedra-incibe.numa.host>

Credenciales en correo de bienvenida

Plataformas recomendadas

<https://open.kattis.com/>

<https://acceptaelreto.com/>

<https://codeforces.com/>

<https://onlinejudge.org/>

<https://www.spoj.com/>

<https://aer.lluiscab.net/institution/universidad-de-cantabria>

<https://open.kattis.com/countries/ESP>

Certificados

Certificado de superación: Asistencia a 4 clases + participación.

Certificado de asistencia: En otro caso que se demuestre participación.

¿Cómo interactuar?

Levantamos manos de teams.

Las sesiones se grabarán, por lo que tendremos el grupo de Telegram disponible para dudas.

El propósito es aprender, no solo competir.

Y responder al correo en copia a todossssss

Experiencia de los Docentes

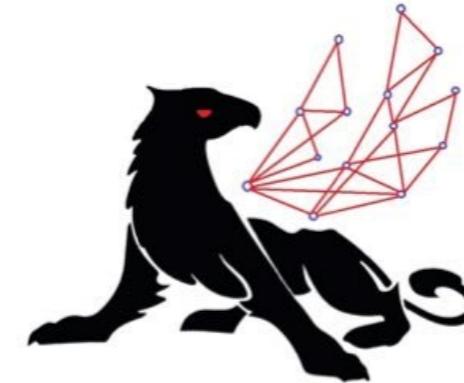


International Collegiate Programming Contest // 2024-2025
The 2024 ICPC Southwestern Europe Regional Contest



The 2024 ICPC Southwestern Europe Regional Contest

SWERC 2024, from November 29th to December 1st, 2024, Lisbon (Portugal), Lyon (France) and Pisa (Italy)



Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

Módulo I: Introducción

Índice del módulo

1. Objetivos del curso.
2. ¿Qué es un juez? ¿Cómo funcionan? ¿Qué jueces existen?
3. ¿Qué competiciones existen? ¿Podemos participar?
4. ¿En qué consiste un problema? ¿Cómo subir un problema?
5. ¡Primer concurso!

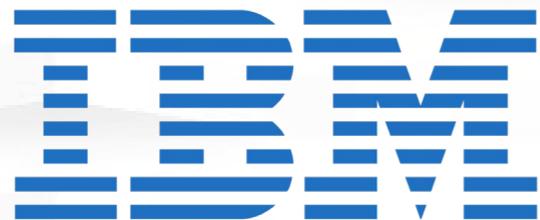
Objetivos del curso

OBJETIVOS

- Mejorar en diseño de algoritmos
 - Elegir de forma más sencilla la estructura de datos
 - Nociones de complejidad
 - Mejorar el ranking de la UC :-)
 - ¡Incluso facilidad para aprobar asignaturas!
-
- Resolver los problemas en el menor tiempo posible
 - Trabajo en equipo (nombres creativos)
 - Representar tu institución, país...

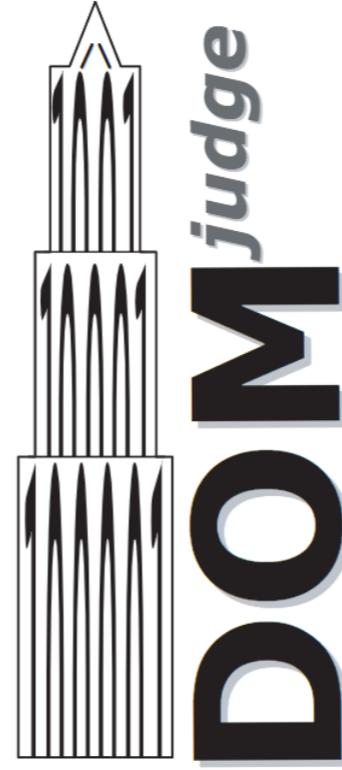
EMPRESAS CAZATALENTOS

- Practicar para futuras entrevistas de trabajo
- Concursos internos
- Cazataleentos

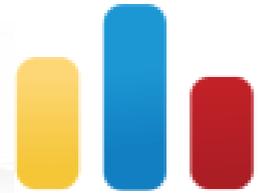


¿Qué es un
Juez?

ALGUNOS JUECES



💡 ¡Acepta el reto!



CODEFORCES
Sponsored by TON

¿Qué competencias
existen?

TIPOS DE COMPETICIONES



- ACM-ICPC:
 - 5 horas de duración
 - Equipos: 3 personas (1 ordenador)
 - Puntuación: problemas resueltos (0/1)
Empates: tiempo + penalizaciones

TIPOS DE COMPETICIONES

ICPC Southwestern Europe Regional Contest (SWERC) 2024

final standings

Filter

RANK	TEAM	SCORE	A	B	C	D	E	F	G	H	I	J	K	L	M
1	Xepelin Ecole Polytechnique	10 1432		46 1 try		18 1 try	120 2 tries		35 1 try	6 1 try	10 1 try	251 2 tries	298 9 tries	295 8 tries	13 1 try
2	flag[10] Università di Pisa	9 839		45 1 try	296 2 tries	21 1 try	116 2 tries		25 1 try	5 1 try	29 1 try		6 tries	192 4 tries	10 1 try
3	UXT Ecole Polytechnique	9 905		51 1 try	241 2 tries	29 1 try	144 1 try	244 2 tries	56 3 tries	4 1 try	14 1 try	2 tries	2 tries		22 2 tries
4	morETHanusual ETH Zürich	9 1046	4 tries	196 3 tries	212 2 tries	26 1 try	256 2 tries		12 1 try	14 1 try	34 1 try			209 1 try	7 1 try
5	TempName Tel Aviv University	9 1164	286 5 tries	94 2 tries	3 tries	33 2 tries	180 3 tries	232 3 tries	31 3 tries	17 1 try	45 1 try	1 try	1 try	3 tries	6 1 try
6	Error-Prone Function Lovers Ecole Polytechnique Fédérale de Lausanne	8 691		43 1 try	242 8 tries	26 1 try	160 2 tries		31 1 try	4 1 try	17 1 try			1 try	8 1 try
7	UPC-1 Universitat Politècnica de Catalunya	8 740		41 1 try	9 tries	16 1 try	256 7 tries	204 4 tries	25 1 try	3 1 try	10 1 try				5 1 try
8	forETHought ETH Zürich	8 817		192 3 tries	163 4 tries	39 2 tries	202 1 try		21 1 try	6 1 try	31 1 try		1 try	3 tries	23 2 tries
9	ENS Ulm 1 École Normale Supérieure de Paris	8 855		65 2 tries	1 try	35 1 try	274 6 tries	1 try	33 1 try	8 1 try	16 1 try	254 3 tries			10 1 try
10	Schwebler's Peacocks The Open University of Israel	8 871		186 4 tries	11 tries	26 1 try	249 4 tries		22 1 try	11 1 try	9 1 try			195 3 tries	13 1 try

<https://swerc.vps.tecnico.ulisboa.pt/domjudge/public>

Raúl Martín Santamaría
Isaac Lozano Osorio
Sergio Pérez Peló

5

TIPOS DE COMPETICIONES

- ACM-ICPC (Proceso de selección)
 - Eliminatorias en la **universidad** si hay más de tres equipos
 - Eliminatorias en el conjunto de **países** que forman una región (South-Western Europe)
 - Eliminatorias entre los potenciales candidatos en todo el **continente** (Super regional europeo (Beta))
 - **Final Mundial**

TIPOS DE COMPETICIONES

- [Codeforces](#) y [Topcoder](#)
 - Concursos muy rápidos y frecuentes
 - Libre para cualquiera
 - Tres o cuatro **divisiones** para novatos y expertos
 - De 95 a 120 minutos de duración
 - Puedes ver y 'romper' el código de otros
 - Sistema de puntuación (mientras más tardes en resolver problemas, más te penalizan en puntos)

TIPOS DE COMPETICIONES

- USACO/COCI/IOI
 - Concursos dirigidos a alumnos de bachiller/secundaria
 - ¡NO SON TAN FÁCILES!
 - Son evaluados con sistemas de puntuación (no binario ni penalizando tiempo de solución)
 - Resultados después de la competición
 - Funcionan por temporadas (de noviembre a abril) por ser eliminatorias para el IOI (International Olympiads in Informatics)

CONCURSOS

- **AdaByron '25 Online** ¿? (<https://ada-byron.es/2025/>)
- **AdaByron Nacional** – 4 y 5 de julio
(<https://ada-byron.es/2025/nac/>)
- **12 Uvas** -> 31/12 (<https://las12uvas.es/>)

Los concursos oficiales FUERZAN que sean **3 personas**, en caso de ser menos o más no permiten participar.

Es buena idea buscar compañeros desde un principio.

¿Cómo subir un
problema?

CARACTERÍSTICAS DE UN PROBLEMA

Enunciado: Se explica el problema con una narración que lo justifica

Análisis del Problema: Se requiere una solución determinista para el problema (siempre encontraremos una solución óptima y válida)

Entrada: Se especifica lo que nuestro programa debe leer

Salida: Se especifica lo que nuestro programa debe mostrar

Ejemplos I/O: Muestras de entrada/salida con el comportamiento esperado para el programa

Límites [Opcionales]: Lo máximo ó mínimo en cuanto a variables que nuestro programa debe tomar en cuenta

CARACTERÍSTICAS DE UN PROBLEMA

- Tipos de Lectura:
 - Un caso: Se lee un caso de prueba y a partir de la entrada se genera una salida y termina la ejecución
 - Múltiples casos: Se leen varios casos de pruebas y, dadas múltiples entradas, se generan múltiples salidas
- No hace falta guardar todos los resultados y mostrarlos al final
- ¡Cuidado con reutilizar estructuras de datos!

¡Primeros problemas!

Lectura de 1 solo caso

```
▶ if __name__ == '__main__':  
    name = input()  
    age = int(input())
```

<https://open.kattis.com/problems/hiphiphurra>

¡Primeros problemas!

Se recibe un entero T y luego vendrán T casos de prueba

```
n = int(input())  
for i in range(n):  
    #- leer datos de cada caso  
    #- codigo + generar salida
```

¡Primeros problemas!

Se leen los casos hasta leer la marca EOF (End-Of-File)

```
import sys

for line in sys.stdin:
    #codigo
```

- `line.split()` para separar por espacios y acceder con `map(int,line.split())` o `[posicion]`.
- `line.strip()` para quitar el salto de línea.

¡Primeros problemas!

Se lee el número de casos hasta que se consiga una condición de parada (generalmente cuando la entrada sea 0).

```
import sys

for line in sys.stdin:
    if(line=='0\n'):
        sys.exit()
    #codigo
```

Posibles veredictos

AC



Tu solución es al menos tan buena como la esperada

- Imprimes correctamente todos los casos ocultos.
- Tu código tarda menos en ejecutarse que el tiempo límite.
- Tu código termina de ejecutarse sin problema.

Posibles veredictos

WA



WRONG ANSWER

La solución que imprimes en algún caso no coincide con la esperada.

Posibles veredictos

RTE



RUN TIME ERROR

Tu código muere por algún fallo en tiempo de ejecución.

- Dividir entre cero.
- Acceder a zonas de memoria no reservada.
- Usar objetos en "null".
- Utilizar librerías externas (Python).

Posibles veredictos

TLE

TIME LIMIT EXCEPTION

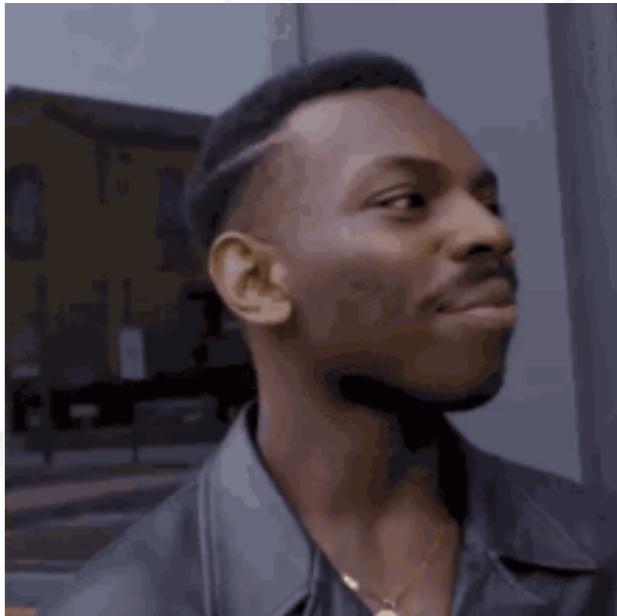
Tu código tarda demasiado en ejecutarse.



- Hay que reducir la complejidad del algoritmo!!!
- Puede esconder otros veredictos.
- Seguramente no es la manera adecuada de resolverlo.

Posibles veredictos

MLE



MEMORY LIMIT EXCEPTION

Tu código utiliza más memoria de la reservada.

- Hay que reducir la memoria del algoritmo!!!
- Puede esconder otros veredictos.

Posibles veredictos

PE



PRESENTATION ERROR

Tu código está bien, pero no lo has impreso de la manera correcta.

- No siempre se contempla, en algunas ocasiones devolverá un WA.

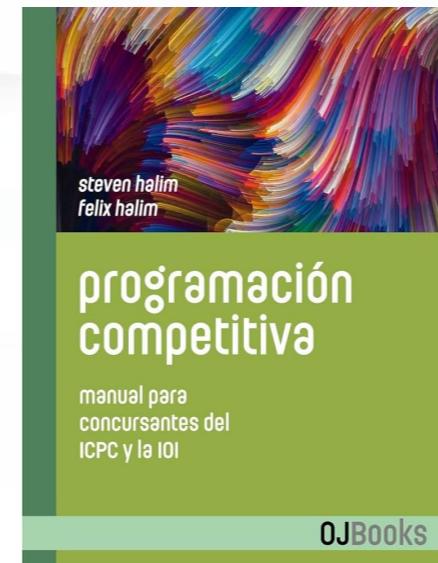
Primer concurso

TENGO INTERÉS ¿CÓMO MEJORAR?

- Practicar lo máximo posible.
- Concursos de Codeforces (semanales).
- Leer, entender y practicar cada tema del libro Competitive Programming 3 o 4.

HERRAMIENTAS

- <https://ideone.com/>
- <https://open.kattis.com/countries/ESP>
- <https://www.udebug.com/>
- <https://www.diffchecker.com/>
- <http://geeksforgeeks.org/>
- <https://aer.lluiscab.net/>



#CátedrasCiber

Módulo I: Introducción