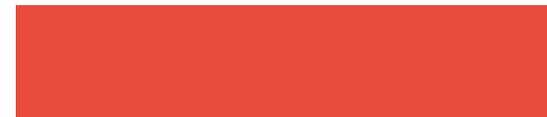


CURSO DE PROGRAMACIÓN COMPETITIVA URJC - 2018



CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2018

Curso:

- 9 sesiones
- 26 de enero - 23 marzo (inclusive)
- Viernes: 17:00 - 19:00
- RAC(?)

Organizadores:

- David Morán (ddavidmorang@gmail.com)
- Sergio Pérez (sergioperezp1995@gmail.com)
- Juan Quintana (juandavid.quintana@urjc.es)

CURSO DE PROGRAMACIÓN COMPETITIVA

URJC - 2018

Fechas importantes:

- Festivo: viernes 02 de febrero (sin sesión)
- Google Hashcode: jueves 01 de marzo
- Ada Byron: viernes y sábado 20-21 de abril
- SWERC 2018 (octubre-noviembre?)
 - Clasificación equipos URJC

Motivación

- Participación en concursos
 - SWERC '18 en París: 3 equipos(?)
 - AdaByron
 - Google Hashcode
 - 12 Uvas
 - ... premios y honor!!!

Motivación

- Empresas patrocinadoras
 - cazatalentos
 - concursos internos
 - entrevistas de trabajo

Motivación

- Programming skills
 - diseño de algoritmos
 - estructuras de datos
 - nociones de complejidad
 - ...aprobar asignaturas!!! (ED, EDA, DAA, ...)



OBJETIVOS EN PROGRAMACIÓN COMPETITIVA

- Resolver los problemas en el menor tiempo posible
- Tener nociones intuitivas:
 - Tipos de problemas, algoritmos...
 - Complejidad vs límite de tiempo (eficiencia)
 - Estructuras de datos necesarias
- Trabajo en equipo (nombres creativos)
- Representar tu institución, país...

PLANIFICACIÓN DEL CURSO

- Semana 1: Introducción
- Semana 2: Estructuras de Datos, Complejidad en Tiempo y espacio básica, algoritmos de ordenamiento y búsqueda
- Semana 3: Algoritmos Voraces, Grafos
- Semana 4: +Grafos, Programación Dinámica
- Semana 5: +Programación Dinámica

PLANIFICACIÓN DEL CURSO

- Semana 6: Estructuras de Datos Avanzadas
- Semana 7: Strings y Algoritmos de flujo de redes
- Semana 8: Geometría
- Semana 9: Mates y teoría de números

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

| RANK | TEAM | SCORE | A  | B  | C  | D  | E  | F  | G  | H  | I  | J  | K  |
|------|--|--------|---|---|---|---|---|---|---|---|---|---|---|
| 1 |  ENS Ulm 1 École Normale Supérieure de Paris | 9 1124 | 1/10 | 1/238 | 1/96 | 1/278 | 2/56 | 1/11 | 2 | 3 | 4/234 | 2/14 | 3/47 |
| 2 |  rETHink togetHER ETH Zürich | 9 1170 | 2/9 | 0 | 1/100 | 2/271 | 1/77 | 1/13 | 1/224 | 0 | 4/195 | 2/29 | 2/112 |
| 3 |  SNS Pisa Scuola Normale Superiore | 7 485 | 1/9 | 1 | 2/100 | 0 | 1/45 | 1/11 | 0 | 2 | 2/166 | 1/15 | 1/99 |
| 4 |  UPC 1 Universitat Politècnica de Catalunya | 7 562 | 1/21 | 2/246 | 3/126 | 0 | 1/56 | 1/4 | 1 | 6 | 1 | 1/12 | 1/37 |
| 5 |  illUSion Università della Svizzera italiana | 7 619 | 1/16 | 1/260 | 1/50 | 0 | 1/82 | 1/9 | 1 | 0 | 0 | 1/21 | 3/141 |
| 6 |  Rubber Duck Forces Technion - Israel Institute of Technology | 7 824 | 1/26 | 0 | 1/199 | 0 | 1/145 | 1/11 | 3/214 | 0 | 0 | 1/15 | 1/174 |
| 7 |  UPC 3 Universitat Politècnica de Catalunya | 6 435 | 1/33 | 1 | 2/186 | 0 | 1/31 | 1/6 | 1 | 0 | 0 | 1/12 | 3/107 |
| 8 |  PDT Politecnico di Torino | 6 454 | 2/16 | 5 | 3/71 | 1 | 2/107 | 1/11 | 0 | 0 | 0 | 1/27 | 1/142 |

TIPOS DE COMPETICIONES

- ACM-ICPC (Proceso de selección)
 - Eliminatorias en la universidad si hay más de tres equipos
 - Eliminatorias en el país si hay más de un número determinado de equipos
 - Eliminatorias en el conjunto de países que forman una región (South-Western Europe)
 - Final Mundial

TIPOS DE COMPETICIONES

- Codeforces y Topcoder
 - Concursos muy rápidos y frecuentes
 - Libre para cualquiera
 - Dos 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

- Facebook Hacker Cup y Google Code Jam
 - Evento de gente masiva online
 - Al menos 4 rondas
 - Suele haber ronda de clasificación, 2 rondas de filtro y luego la fase final
 - Dos tipos de evaluación (small y large)
 - El caso small se corrige automáticamente
 - El caso large se corrige al terminar la competición
 - Se permite cualquier tipo de solución (incluso manual ó *hardcodeada*) que permita llegar al output

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

CARACTERÍSTICAS DE UN PROBLEMA

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

Análisis del Problema: Se requiere una solución determinística 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
- ¡Cuidado con reutilizar estructuras!
- No hace falta guardar todos los resultados y mostrarlos al final

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA DE T CASOS

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

```
Scanner l = new Scanner(System.in);
int n = l.nextInt();
l.nextLine();
for (int i = 0; i < n; i++) {
    //Código del programa
}
```

```
int t; scanf("%d",&t);
for(int i=0;i<t;i++){
    // código del programa
}
```

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA HASTA EOF

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

```
Scanner l = new Scanner(System.in);
String n;

while(l.hasNextLine()){
    n = l.nextLine();
    //Código del programa
}
```

```
int n;
while(scanf("%d",&n)!=EOF){
    // código del programa
}
```

CARACTERÍSTICAS DE UN PROBLEMA: LECTURA HASTA CASO EN 0

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

```
Scanner l = new Scanner(System.in);
int n = l.nextInt();
l.nextLine();
while(n!=0){
    //Código del programa
}
```

```
int n;
while(scanf("%d",&n)!=EOF && n!=0){
    // código del programa
}
```

LENGUAJES DE PROGRAMACIÓN

- C
- C++
- Java
- Python (muy nuevo)

Ejemplo Problema: Último Dígito del Factorial

<https://www.aceptaelreto.com/problem/statement.php?id=114>

¡Acepta el reto!

Problemas

Estadísticas

Documentación

Login ▾

Buscar ▾

Estás en: Inicio

¿Qué es?

¡Acepta el reto! es un almacén y juez en línea de problemas de programación en español que acepta soluciones en C, C++ y Java.

No es un mero listado de problemas, sino mucho más. *¡Es un corrector automático!*

Si quieres poner a prueba tu habilidad programando y compararla con la de otros, ¡éste es tu sitio!

¿Por dónde empiezo?

Lo primero que querrás hacer será leer algunos de los múltiples [problemas](#) disponibles. Si no sabes por cuál empezar, puedes recorrer las diferentes [categorías](#) o mirar el *problema de la semana* que te proponemos abajo.

Si te llama la atención algún problema, crees que eres capaz de resolverlo y quieres intentarlo, [regístrate](#). ¡Es fácil, rápido y no te enviaremos spam! Con tu cuenta, podrás enviar tus soluciones y compararlas con las de [otros usuarios](#).

¿Aceptas el reto?

Ejemplo Problema: Dividir Factoriales

<https://www.aceptaelreto.com/problem/statement.php?id=190>

¡Acepta el reto! Problemas Estadísticas Documentación

Login Buscar

Estás en: Inicio

¿Qué es?

¡Acepta el reto! es un almacén y juez en línea de problemas de programación en español que acepta soluciones en C, C++ y Java.

No es un mero listado de problemas, sino mucho más. ¡Es un corrector automático!

Si quieres poner a prueba tu habilidad programando y compararla con la de otros, ¡éste es tu sitio!

¿Por dónde empiezo?

Lo primero que querrás hacer será leer algunos de los múltiples problemas disponibles. Si no sabes por cuál empezar, puedes recorrer las diferentes categorías o mirar el *problema de la semana* que te proponemos abajo.

Si te llama la atención algún problema, crees que eres capaz de resolverlo y quieres intentarlo, [regístrate](#). ¡Es fácil, rápido y no te enviaremos spam! Con tu cuenta, podrás enviar tus soluciones y compararlas con las de otros usuarios.

¿Aceptas el reto?

Registrarse en el entrenamiento

<http://programming-urjc.herokuapp.com/>

https://docs.google.com/forms/d/e/1FAIpQLScUulmEQB3HAYjOVEN_n3iOhllrq41XxRpZeMUfN2RiwctElA/viewform

Semana que viene

- Estructuras de Datos, Complejidad y Algoritmos de búsqueda y ordenamiento
- Intentar familiarizarse con SPOJ y AceptaElReto

Hasta la vista

Baby

