Programación Competitiva

B.S. Rodolfo Mercado Gonzales

Programación Competitiva

Consiste en resolver problemas “lo antes posible”, a través de programas computacionales, minimizando los siguientes recursos :

- ❏ Tiempo de ejecución del programa
- ❏ Memoria usada por el programa

Programación Competitiva

“El objetivo es formar informáticos que estén preparados para producir mejor software y enfrentarse en el futuro a problemas de investigación.”

Fundadores ACM-ICPC

Tiempo de Ejecución

- Es el tiempo que le toma a un algoritmo procesar una determinada entrada.
- En las competencias hay un límite para el tiempo de ejecución que generalmente es **1 segundo.**

Time limit exceeded on test 12	2000 ms	16200 KB

Memoria

- Las variables usadas en un algoritmo son las que ocupan memoria.
- En las competencias hay un límite para el uso de memoria, generalmente **256 MB.**

Memory limit exceeded on test 1	93 ms	262100 KB

¿Cómo son los problemas?

- Nos enfrentamos a problemas algorítmicos, que previamente han sido resueltos al menos por el autor.
- No son imposibles de resolver.

¿Cómo son los problemas?

Enunciados en inglés, con la siguiente estructura :

- ❏ Restricciones del tiempo de ejecución y memoria.
- ❏ Descripción del problema (statement).
- ❏ Descripción de la entrada y salida (input, output)
- ❏ Restricciones de los datos de entrada (constraints).
- ❏ Ejemplo de entrada y salida (sample input, sample output).

¿Cómo son los problemas?

A. Nearest Minimums

time limit per test: 2 seconds
memory limit per test: 256 megabytes
input: standard input
output: standard output

You are given an array of n integer numbers $a_0, a_1, ..., a_{n-1}$. Find the distance between two closest (nearest) minimums in it. It is guaranteed that in the array a minimum occurs at least two times.

Input

The first line contains positive integer n ($2 \le n \le 10^5$) — size of the given array. The second line contains n integers $a_0, a_1, ..., a_{n-1}$ ($1 \le a_i \le 10^9$) — elements of the array. It is guaranteed that in the array a minimum occurs at least two times.

Output

Print the only number — distance between two nearest minimums in the array.

Examples

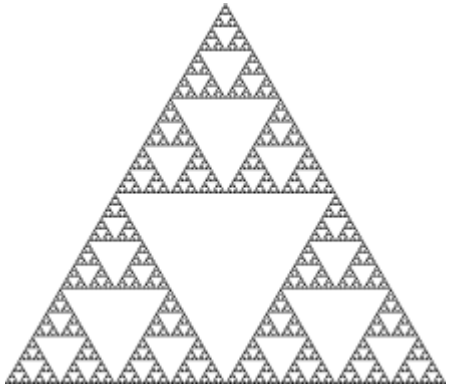
input
2 3 3
output
1

¿Qué conocimientos debo tener/adquirir?

- Razonamiento lógico-matemático.
- **Algoritmos y estructuras de datos**.
- Un lenguaje de programación a nivel básico.

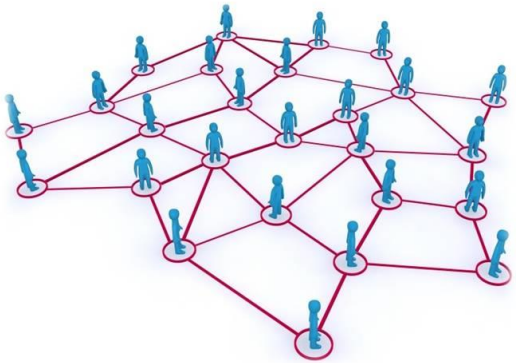
Algoritmos y Estructuras de Datos

- Análisis de Algoritmos
- Standard Template Library
- Fuerza Bruta
- Recursividad
- Backtracking
- Divide y Vencerás
- Búsqueda Binaria
- Programación Dinámica



Algoritmos y Estructuras de Datos

- Teoría de números
- Geometría Computacional
- Grafos
- Estructuras de Datos Avanzadas
- Procesamiento de Cadenas
- Teoría de Juegos



Competencias de Programación

- Las competencias consisten en resolver un conjunto de problemas algorítmicos en un determinado tiempo.

- Gana quien resuelva la mayor cantidad de problemas, en caso de empate quien lo hizo en menor tiempo será el vencedor.

Evaluación de una solución

- Se envía el código fuente a un juez online.

Evaluación de una solución

- Este juez se encargará de testear el programa con un input secreto.

Who	Problem	Lang	Verdict
system_1	911D - Inversion Counting	GNU C++	Running on test 1
army_of_one	459A - Pashmak and Garden	GNU C++11	Running on test 6
manojkannekanti	911A - Nearest Minimums	GNU C++14	Running on test 4

Evaluación de una solución

- En unos segundos el juez dará un veredicto para tu solución.

  AC (accepted) – Tu solución es correcta.

  WA (wrong answer) – Tu programa da una respuesta incorrecta.

  TLE (time limit exceeded) – Tu programa tarda mucho tiempo.

  RE (runtime error) – Tu programa se cae durante su ejecución.

  CE (compilation error) – Tu programa no compila.

Concursos en Equipo

ACM-ICPC

- Competencia organizada por la ACM y auspiciada por IBM.
- Compiten equipos de 3 estudiantes.
- Consta de 2 fases, cada una de 5 horas :

- **Regional Sudamérica/Sur**: Argentina, Perú, Chile, Bolivia, Uruguay y Paraguay (3 cupos para el mundial).

- **Final Mundial** : todos los equipos clasificados de las distintas regionales.

ACM-ICPC

ACM-ICPC


2012 World Finals
acm International Collegiate Programming Contest
IBM
event sponsor
ICPC 2012
University of Warsaw
HOST
IBM
IBM
IBM
IBM


Welcome
PARTICIPANTS
acm
IBM

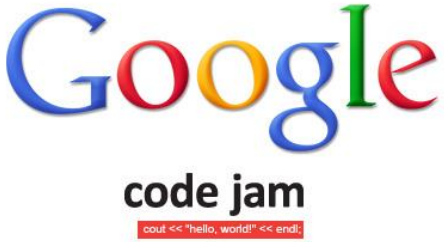

Welcome
PARTICIPANTS
2016 World Fin
Phuket
acm
IBM
Universidad Nacional de Ingeniería - FIIS
2016 ACM-ICPC World Finals

IEEExtreme

- Competencia organizada por la IEEE.
- Compiten equipos de 3 estudiantes.
- Consta de una sola fase.
- El concurso dura 24 horas.

IEEExtreme

Concursos Individuales



Beneficios de Competir

1. Dominarás algoritmos y estructuras de datos.
2. Mejorarás tus habilidades para la resolución de problemas.
3. Aprendes a trabajar en equipo y bajo presión.
4. Aprenderás C++ y Python.

```cpp
#include <bits/stdc++.h>
using namespace std;

struct Tuple{
    int gcd, x, y;
    Tuple( int a, int b, int c ){
        gcd = a, x = b, y = c;
    }
};

Tuple extGcd( int a, int b ){
    if ( b == 0 ) return Tuple( a, 1, 0 );
    Tuple ret = extGcd( b, a % b );
    return Tuple( ret.gcd, ret.y, ret.x - a/b * ret.y );
}

int main( ){
    Tuple ans = extGcd( 3, 5 );
    cout << ans.x << " " << ans.y << endl;
}
```

Beneficios de Competir

5. Puedes participar en campamentos de programación.

Argentina

Beneficios de Competir

5. Puedes participar en campamentos de programación.

Perú

Beneficios de Competir

6. Incrementas tus posibilidades de trabajar en las mejores empresas del ámbito tecnológico

Minimum qualifications:

- Currently enrolled in a full-time Bachelor's in Computer Science or related technical field, returning to BA/BS program or enrolled in another full-time degree program after completion of the internship.
- Completed projects (inside or outside of school) or classes focused on Data Structures and Algorithms; experience with algorithms, using data structures to solve problems, and interpreting algorithms and contributing ideas to their development.
- Experience writing code fixes and tools to solve problems in one or more of the following languages: C, C++, Java, JavaScript, or Python (e.g., remove duplicate elements from a list).
- Experience with linear coding and use language features when necessary (e.g., data structures, branching, function-calls, and conditionals).

Beneficios de Competir

6. Incrementas tus posibilidades de trabajar en las mejores empresas del ámbito tecnológico.

Beneficios de Competir

7. Incrementas tus posibilidades de hacer estudios en el extranjero.

Beneficios de Competir

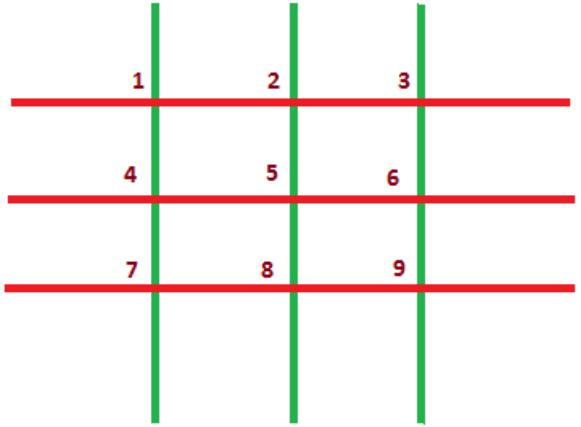
8. Puedes ganar muchos premios.

Nuestro primer problema

Egor y Peter empiezan un juego sobre una grilla compuesta de n varillas horizontales y m verticales.

Ambos juegan en turnos, siendo Egor el que empieza. Durante su turno, el jugador debe escoger un punto de intersección de la grilla y retirar todas aquellas varillas que pasen por dicho punto.

Un jugador perderá si es que no puede hacer movimientos (ya no hay puntos de intersección).



Nuestro primer problema

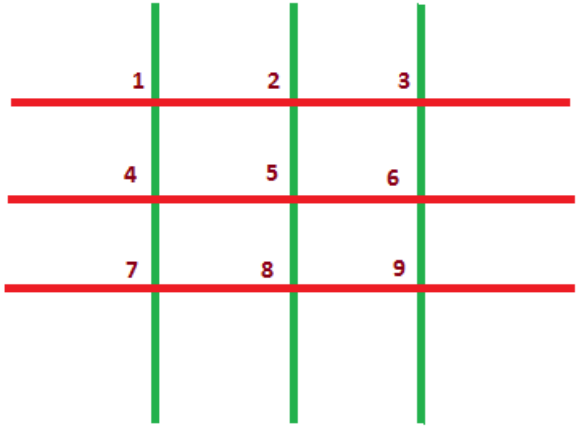
Asumir que ambos juegan óptimamente (buscan ganar). ¿Quién ganará el juego?

Entrada
Se le darán los enteros n y m.

Salida
Indicar el nombre del ganador(Egor o Peter)

Entrada Ejemplo
2 2

Salida Ejemplo
Peter



Fuente: codeforces – game with sticks

¿Cómo afrontar un problema?

MÉTODO PÓLYA

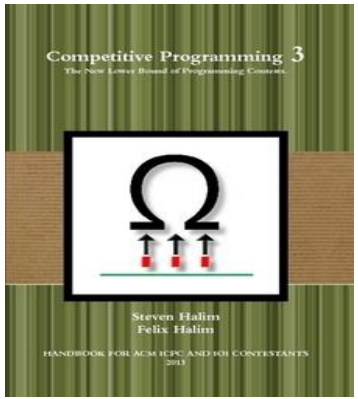
- **Entender el problema :** identificar los datos de entrada, salida y las restricciones.
- **Diseñar un plan:** recordar problemas similares, resolver el problema reduciendo restricciones (caso particulares), resolver casos generales, intentar con los algoritmos que conocemos, ir desde la solución “más ingenua” hasta la buscada.
- **Ejecutar el plan:** implementar nuestro algoritmo (si encontramos alguna dificultad es posible cambiar de estrategia).
- **Examinar la solución:** verificamos los pasos anteriores y testeamos nuestro programa.

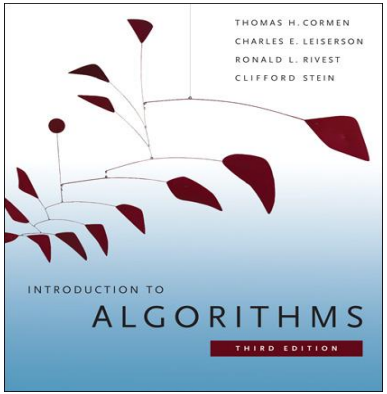
Tips para ser competitivo

- ❑ **Tipear rápido** (typingtest).
- ❑ Identificar rápidamente el tipo de problema.
- ❑ Hacer análisis algorítmico.
- ❑ Dominar tu lenguaje de programación preferido.
- ❑ Testear tu solución.
- ❑ Practicar a full.
- ❑ Trabajo en equipo

Referencias

- Halim, Steven and Halim, Felix. Competitive Programming 3.
- Cormen, Thomas H. Introduction to Algorithms.





¡ Good luck and have fun !