



# TEORÍA DE NÚMEROS: REGLAS DE DIVISIBILIDAD. AUTÓMATA DE DIVISIBILIDAD

**Autores:**

Colectivo de Entrenadores ACM-ICPC UM

Marzo / 2019

ENTRENAMIENTO PARA CONCURSANTES  
ACM-ICPC DE LA UNIVERSIDAD DE  
MATANZAS

# Objetivos

- ▶ Familiarizar a los estudiantes con conceptos de Teoría de Números.
- ▶ Caracterizar las reglas de divisibilidad.
- ▶ Caracterizar un autómata de divisibilidad.

# Objetivos

- ▶ Implementar el algoritmo autómata de divisibilidad.
- ▶ Identificar problemas donde la solución sea aplicando las reglas de divisibilidad.
- ▶ Identificar problemas donde la solución sea aplicando un autómata de divisibilidad.

# Bibliografía

- ▶ *Manual de preparación para concursantes ACM-ICPC de la Universidad de Matanzas.* Publicado en el Entorno Virtual de Aprendizaje de la Universidad de Matanzas ( [eva.umcc.cu](http://eva.umcc.cu) ). Sección *Bibliografía*.

# Teoría de números

La teoría de números es la rama de las matemáticas que estudia las propiedades de los números, en particular los enteros.

- ▶ Números primos.
- ▶ Reglas de divisibilidad.
- ▶ Teorema del resto chino.

# Problema a resolver

Sean  $A$  y  $B$  números enteros se desea saber si  $A$  es divisible por  $B$ .  $A \bmod B == 0$

Muy trivial verdad ?

# Problema a resolver

Pero si  $A$  puede ser un valor de hasta  $10^{10^6}$  y  $B$  un valor que si puede ser almacenado en un tipo de dato numérico entero.

# Reglas de divisibilidad

Las reglas de divisibilidad nos permiten saber, de forma más o menos rápida, si un número es divisible entre otro sin la necesidad de dividir.



# Reglas de divisibilidad

Alguna de las reglas de divisibilidad conocidas son:

- ▶ **2:** Si termina en 0, 2, 4, 6 ó 8.
- ▶ **3:** Si la suma de sus cifras es múltiplo de 3.
- ▶ **4:** Si sus dos últimas cifras son 00 ó un múltiplo de 4 (12, 16, 20, 24, 28, 32, 36 y 40).

# Reglas de divisibilidad

Alguna de las reglas de divisibilidad conocidas son:

- ▶ **5:** Si termina en 0 y 5.
- ▶ **7:** Cuando la diferencia entre el número sin la cifra de las unidades y el doble de la cifra de las unidades es 0 ó múltiplo de 7.
- ▶ **8:** Si sus tres últimas cifras son ceros o múltiplo de 8.

# Reglas de divisibilidad

Alguna de las reglas de divisibilidad conocidas son:

- ▶ **9:** Si la suma de sus dígitos nos da múltiplo de 9.
- ▶ **10:** Si la cifra de las unidades es 0.
- ▶ **11:** Si la suma de las cifras que ocupan un lugar par menos la suma de las otras cifras es 0 ó un múltiplo de 11 (11, 22, 33, 44,...)

# Reglas de divisibilidad

Alguna de las reglas de divisibilidad conocidas son:

- ▶ **25:** Si sus dos últimas cifras son ceros o múltiplo de 25.

# Solución de problema

Existen dos variantes una usando las ciertas ventajas que nos puede dar el lenguaje de programación Java y la otra aplicando un poco de análisis matemático.

# Solución: Java

El lenguaje de programación de Java nos ofrece la clase *BigInteger* para operar y manipular números grandes como lo pueden ser los que hemos mencionado anteriormente.

# Solución: Java

```
import java.math.BigInteger;
import java.util.Scanner;

public class Main {
    public static void main(String[] args) {
        Scanner in=new Scanner(System.in);
        BigInteger a=in.nextBigInteger();
        BigInteger b=new BigInteger("495");
        if(a.mod(b).compareTo(BigInteger.ZERO)==0)
            System.out.println("Es divisible");
        else
            System.out.println("No es divisible");
    }
}
```

# Solución: Java

Esta variante es bastante simple, fácil de implementar pero importante tiene en la mayoría de los casos un alto consumo de memoria.



# Solución: Análisis matemático

Esta variante es más efeciente aunque tiene un grado análisis mayor que el anterior. Es la variante que deberían utilizar aquellos amantes de C++.

# Solución: Análisis matemático

- ▶ El primer paso es almacenar el número  $A$  en una variable de tipo cadena *string*
- ▶ El segundo paso es saber quien es  $B$  si es un valor fijo siempre o es una valor variable para cada caso.

# Solución: Análisis matemático

- ▶ El primer paso es almacenar el número  $A$  en una variable de tipo cadena *string*
- ▶ El segundo paso es saber quien es  $B$  si es un valor fijo siempre o es una valor variable para cada caso.

# Solución: Análisis matemático

Valor  $B$  fijo y conocido

- ▶ Analizamos si el valor  $B$  es un valor que tiene definida su regla de divisibilidad.
- ▶ De ser así implementamos un algoritmo que compruebe la regla de divisibilidad del valor  $B$  en la cadena donde esta almacenado el valor  $A$

# Solución: Análisis matemático

Valor  $B$  fijo y conocido

- ▶ Analizamos si el valor  $B$  es un valor que no tiene definida su regla de divisibilidad.
- ▶ Lo descomponemos en factores (primos preferentemente) que tengan definidas sus reglas de divisibilidad y luego solo debemos comprobar que la cadena cumpla con cada una de las reglas de divisibilidad de cada factor.

# Solución: Análisis matemático

Valor  $B$  variable y desconocido

- ▶ Aplicar un autómata de divisibilidad.
- ▶ Un autómata de divisibilidad permite saber si un número  $X$  en una base  $Y$  es divisible por un número  $Z$  el cual esta en base decimal.

# Autómata

Un autómata es un modelo matemático para una máquina de estado finito (FSM sus siglas en inglés). Una FSM es una máquina que, dada una entrada de símbolos, *salta* a través de una serie de estados de acuerdo a una función de transición.

# Autómata de divisibilidad

Un autómata de divisibilidad va tener una cantidad de estados igual a  $N$ , donde cada estado se enumera de 0 a  $N-1$ .



# Autómata de divisibilidad

El estado inicial es 0. Si después de analizar la cadena por el autómata el estado final es 0 nuevamente se puede expresar que la cadena es divisible por  $N$ .

Un autómata de divisibilidad realiza una simulación de la división por galera.

# Autómata de divisibilidad

```
bool automatDivisible(string _number,int _base, int
    _decimal){
    int state=0;
    int size=_number.size();
    for(int i=0;i<size;i++)
        state=(state*_base+value(_number[i]))%
            _decimal;
    if(!state)return true;
    else return false;
}
```

# Problemas

Problemas publicados en AOJ cuya solución se basa en aplicar reglas de divisibilidad o autómatata de divisibilidad.

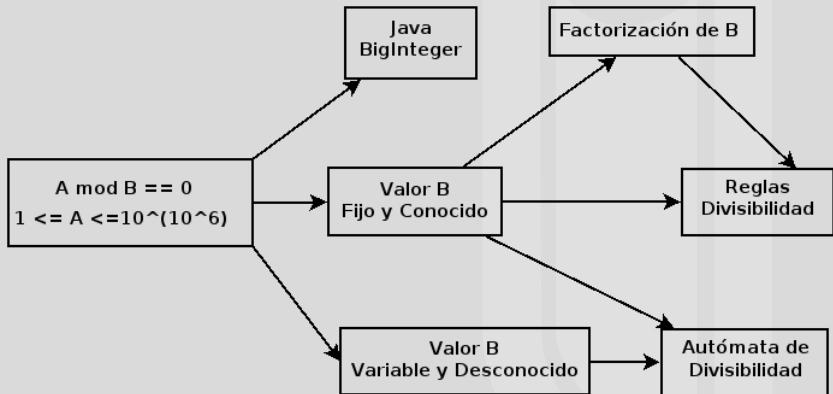
- ▶ You Can Say 11
- ▶ Div 5
- ▶ Div 4
- ▶ Div 3

# Problemas

Problemas publicados en AOJ cuya solución se basa en aplicar reglas de divisibilidad o autómatas de divisibilidad.

- ▶ Div 6
- ▶ Divisibility 495
- ▶ LOJ - Large division

# Conclusiones



# UNIVERSIDAD DE MATANZAS

cosechando el saber

FIN