SEMILLERO de PROGRAMACIÓN

# Solucionario: Nivelación Ronda 4

Temática General: Aritmética modular, entradas grandes y demás

# A



- Nombre: Tarea interminable
- Link:https://trainingcenter.cloud.ufps.edu.co/problemas/63/detalle/es
- Análisis: Dada una operación debemos de dos números debemos mostrar su resultado. (MULTIPLICACION, SUMA, RESTA). Como los valores que pueden tener a y b estan entre 1 y 10^40. Se nos dificulta solucionar este problema en c++, ya que el máximo valor que podemos trabajar en c++ aproximadamente es 10^18. Por lo tanto, tocaría una clase de BigIntegger en c++ para solucionarlo. Por su defecto, podemos solucionar este problema en python o en java fácilmente
- Implementación: Clase BigIntegger de Java o implementarla en c++ o solucionarlo en python.
- Consideraciones: Como python es relativamente lento, se sugiere trabajarlo con un metodo main o las entradas rápidas de stdin y stdout, para evitar TLE o un RTE.

# B



- Nombre: Tarea interminable II
- Link:https://trainingcenter.cloud.ufps.edu.co/problemas/66/detalle/es
- Análisis: Debemos operar N <=10 números, (SUMA, MULTIPLICACION) y mostrar el resultado en %1007. Cada ni <=10^8. En el peor de los casos debemos multiplicar 10 números con un valor de 1e8. Eso nos daría un número de 1e80. Por lo tanto, para evitar los números se agranden mucho, toca aplicar las propiedades de la aritmética modular en el campo de la suma y la resta. Otra solución, sería hacerlo en python y simplemente mostrar el resultado en %1007.
- Implementación: Para una solución en c++, aplicar las propiedades de la aritmética modular.
- Consideraciones: Para python, usar metodos fast i/o para evitar TLE o RTE.

# C



- Nombre: Cancele
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/278/detalle/es
- Análisis: Sumar dos números, lo difícil aquí es que estos pueden tener valores entre 1 y 10^50..
- Implementación: Usar BigIntegger en java o implementarlo en c++ o solucionarlo en python.
- Implementación: Para python, usar métodos fast i/o para evitar TLE o RTE

# D



- Nombre: Problema con un nombre ridículamente largo pero una descripción ridículamente corta
- 
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/12/detalle/es

- Análisis: Debemos hacer la siguiente operación, (66^n) %100. Donde n tiene un valor entre 0 y 10^1000. Para solucionarlo debemos usar la función pow de python, ya que con el operador de potencia normal (**) nos da RTE.

pow( base, exponente, módulo)

- Implementacion: Usar la función pow de python o hacer casos a mano y detectar los resultados cíclicos. Ya que en cierto rango los resultados para 66^n%100 se repiten.

# E



- Nombre: Superbowl

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/149/detalle/en

- Análisis: Nos dan la sumatoria y la diferencia en valor absoluto de dos números. La tarea es hallarlos. Simplemente despejando, encontramos que:

  a = sumatoria – b
  b = absoluto(s-d)/2

  Ahora, debemos verificar que los resultados de a y b, concuerden con la sumatoria y la diferencia dada. Si esto es asi, mostramos a y b, de lo contrario mostramos "impossible“

- Consideraciones: Usar long long, ya que el long al que se refiere el ejercicio es de java.



- Ejemplo:

Sumatoria = 40
|Diferencia| = 20
b = abs(40-20)/2 = 20/2 = 10
a = 40 - 10 = 30

- Ejemplo 2:

Sumatoria = 20
|Diferencia| = 40
b = abs(20-40)/2 = 20/2 = 10
a = 20 - 10 = 10
a + b = 10 + 10 = 20 - > Correcto
|a-b| = |10-10| = 0 - > Incorrecto Para este caso, es impossible

# F



- Nombre: The 3n + 1 problema

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/250/detalle/en

- Análisis: Nos dan un rango de a y b, con cada número de este rango debemos aplicar lo que describe el problema, si i-nesimo número es par divido por dos y si es impar multiplicó por 3 y le sumó 1. Debemos hacer esto repetidamente hasta que i-nesimo número tome el valor de 1, la finalidad de hacer esto, es contar cuantas operaciones nos tomó para que ese número llegue a 1. Entonces, dado el rango, debemos el máximo número de pasos por cada número del rango. Tener en cuenta, que primero debemos determinar si a < b o b< a, para hacerlo de menos a mayor o viceversa.

- Ejemplo:

a = 3 y b = 1
Empezamos desde 1 hasta 3
1, como ya esta en uno, no hago nada
2, es par, lo divido y llevo un paso
Ahora 2 es 1, como ya llegué a uno, no haga nada más.
Ahora el último número del rango.
3 Impar - > 3*3+1 = 10 - >1 paso
10 par - > 10/2 = 5 - >2 pasos
5 impar - > 5*3+1 = 16 - > 3 pasos
16 par - > 16/2 = 8 - > 4 pasos
8 par - > 8/2 = 4 -> 5 pasos
4 par - > 4/2=2 - > 6 pasos
2 par - > 2/2 = 1 - > 7 pasos
1 Me detengo
Entonces, entre el rango de a y b, el número con mayor pasos es 3, que son 7 pasos.

# G



- Nombre: Bisiesto
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/347/detalle/es
- Análisis: Debemos encontrar el primer año bisiesto de cierto mundo, nos dan año actual, la diferencia de años para que haya un año bisiesto y del cual debe ser múltiplo para ser considerado bisiesto, y el número de años bisiestos que han pasado. (1<=D, K<=A<=10^18). Lo primero a tener en cuenta, es si es año actual es bisiesto o no, si es así debemos restar uno en los años bisiestos que han pasado, de lo contrario debemos encontrar año bisiesto y restarle la multiplicacion de los años bisiestos pasados y la diferencia entre cada uno.
- Consideraciones: Como los rangos de los valores pueden tener un valor dw 1e18, debemos usar variables unsigned long long
- Ejemplo:

  2019 4 109
  El próximo  bisiesto es 2020
  Ahora, 2020-(4*109) = 1584

Ejemplo 2:

35520 5 1043
Como 35520 es bisiesto, restamos 1 en los años pasados
35520-(5*1042) = 30310

# H



- Nombre: Números perfectos

- Link:https://trainingcenter.cloud.ufps.edu.co/problemas/376/detalle/es

- Análisis : Determinar si un número es perfecto en base de k. es perfecto en base K si es divisible en K y además tanto el resultado de la suma de sus dígitos como la multiplicación son divisibles en K. Si no cumple ninguna de estas condiciones el número es aburrido. Donde s este entre 100 y 10^1000. La solución en python es muy sencilla, simplemente usar un contador de condiciones y dada el número de condiciones mostrar si es perfecto, claro antes toca hacer lo que pide el problema. La solución es c++ es mucho más compleja, ya qje este número no cabe en ninguna variable, una manera de trabajar este problema es ir codigendo cada carácter (dígito) del número y aplicarle aritmética modular, de esta manera el número se vuelve más pequeño

- Ejemplo:
  K= 3 y s =123
  123%3=0 - >SI → 1*2*3=6%3=0 - > SI → 1+2+3=6%3=0 - > SI
  Este número es perfecto en base de 3



- Nombre: ¡Huracán!
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/301/detalle/es
- Análisis: Nos dan un listado de un par de números, la tarea es mostrarlos en orden ascendente
- Implementación: Crear una estructura de datos tipo set ya que las casos no se repetirán, este set va a contener un pair, luego simplemente mostramos los datos en el set.



# J



- Nombre: Conversación de base

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/62/detalle/es

- Saberes previos: Saber dividir, que son números binarios, decimales y hexadecimales.
- Analisis: Nos dan un numero en cierta base, la tarea es mostrarlo en las dos bases restante siguiendo este orden: Dec-Hex-Bin. Una solución es crear un método que convierta el número a cada base respectiva, puede ayudarse con la función stoi y leer el número como cadena y pasarlo a int en la base decimal.

- Ejemplo:

stoi(cadena,NULL,base)
101 bin
num = stoi(101,0,2) Pasa binario a decimal
Para  binario es 2
Para hexadecimal es 16
Para decimal es 10

Otra solución, es adecuar su método que convierta todas las bases, donde recibe el número y la base a convertir, esto se puede ya que para convertir a binario, se divido por 2 sucesivamente, 16 para hex y 10 para dec.

# k

SEMILLERO de PROGRAMACIÓN

- Nombre: Robot
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/312/detalle/es
- Análisis: Error 404 análisis not found
- Editorial: https://www.hackerrank.com/challenges/robot/problem
- Nota: Aunque este problema sea supuestamente de dificultad 1, la solución es avanzada. Ya que utiliza dp y estructuras de datos tipo arboles para la solución, ya que únicamente con dp da TLE.

# I

- Nombre: Truco de magia
- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/320/detalle/es
- Análisis: La tarea es descubrir la técnica del mago, para encontrar el número en el que piensas entre 0 y 31, dadas 5 imágenes con números y unas preguntas sencillas. ¿Tu número esta en esa imagen?. Lo primero que se da cuenta, es que todas las imágenes tienen números entre el 1 y el 31, por lo tanto si respondes NO en todas las imágenes, tu número es 0. De no ser así, la tarea es ir tachando números, por ejemplo, si tu número no esta en la primera imagen, tachas todos los números que estén en esa imagen y en las otras imágenes, repitiendo esto 5 veces. Luego, te quedan algunos números sin tachar, dadas las imágenes que respondes SI debes encontrar el que aparece en todas.
- Nota: En las 3 primeras imágenes debes agregarle el número 15,ya que este hace falta. Todas las imágenes tienen 16 números, la solución es puro for y condiciones

# M



- Nombre: Los tres hermanos

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/440/detalle/es

- Análisis: Dado n dulces, debemos repartirlos entre tres hermanos. Si n%3==0 entonces, los tres hermanos tienen igual cantidad de dulces, si n%3==1 quiere decir que sobró un dulce, asi que ese que sobró se le da a darla, si n%3==2 entonces sobran dos dulces que se le dan a darla y a David.

- Ejemplo

3 //3 dulces → 1 1 1 //Todos tienen la misma cantidad
4 //4 dulces, sobra 1 → 2 1 1
5 //5 dulces, sobran 2 → 2 2 1

# N



- Nombre: Números maravillosos

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/327/detalle/es

- Análisis: Para determinar si un número es maravilloso de orden x, tenemos que leer una cadena y ir sumando sus dígitos, además de ir comprobando si Npar digitos son pares y Nimpar digitos son impares, pues si tiene uno impar y uno par, ya se considera número completo. Iremos repitiendo este proceso hasta que el número sea menor de 10. Despues verificamos cuantos números completos tenemos, si la cuenta es 0, no es maravilloso, de lo contrario es maravilloso de orden x (x = conteo)

- Implementación: Uso de to_string para ir convirtiendo la suma de digitos en cadena o si gusta, use otra conversación de int a string.

# O



- Nombre: Charlie y la fabrica de Chocolates

- Link: https://trainingcenter.cloud.ufps.edu.co/problemas/65/detalle/es

- Análisis: nos damos cuenta que es una progresión aritmética la tarea, seria hallar el enésimo término, que viene siendo el costo de las cajas para eso multiplicamos la diferencia por el término inicial elavado a n-1
  n = d*a1^n-1

- Implementación: Implementación: Usar un for para hacer las multiplicaciones o usar pow (Tenga en cuenta que pow da un número real, tendría que guardar el resultado en un int para evitar un wa)


SEMILLERO de
PROGRAMACIÓN