

# The $3n + 1$ problem

## Background: (black in english - blue in spanish)

Problems in Computer Science are often classified as belonging to a certain class of problems (e.g., NP, Unsolvable, Recursive). In this problem you will be analyzing a property of an algorithm whose classification is not known for all possible inputs.

## Introducción: (azul en español - negro en inglés)

Los problemas en Ciencias de la Computación a menudo se clasifican como pertenecientes a ciertas clases de problemas (ej.: NP, No resolubles, Recursivos). En este problema ud. analizará una propiedad de un algoritmo cuya clasificación no se conoce para todas las posibles entradas.

## The Problem:

Consider the following algorithm:

1. input  $n$
2. print  $n$
3. if  $n = 1$  then STOP
  4. if  $n$  is odd then  $n = 3n + 1$
  5. else  $n = n / 2$
6. GOTO 2

Given the input 22, the following sequence of numbers will be printed 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

It is conjectured that the algorithm above will terminate (when a 1 is printed) for any integral input value. Despite the simplicity of the algorithm, it is unknown whether this conjecture is true. It has been verified, however, for all integers  $n$  such that  $0 < n < 1,000,000$  (and, in fact, for many more numbers than this.)

Given an input  $n$ , it is possible to determine the number of numbers printed (including the 1). For a given  $n$  this is called the cycle-length of  $n$ . In the example above, the cycle length of 22 is 16.

For any two numbers  $i$  and  $j$  you are to determine the maximum cycle length over all numbers between  $i$  and  $j$ .

## El Problema:

Considere el siguiente algoritmo:

1. input  $n$
2. print  $n$
3. if  $n = 1$  then STOP
  4. if  $n$  is odd then  $n = 3n + 1$

5. else  $n = n / 2$

6. GOTO 2

Para la entrada 22, se imprimirá la siguiente secuencia de números 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Se conjetura que el algoritmo anterior terminará (cuando se imprime un 1) para cualquier valor entero. A pesar de la simplicidad del algoritmo, se desconoce si esta conjetura es verdadera. Ha sido verificado, sin embargo, para todos los enteros tales que  $0 < n < 1,000,000$  (y, de hecho, para muchos enteros más grandes....)

Dado un entero  $n$ , es posible determinar el número de números que serán impresos (incluyendo el 1). Para un  $n$  dado, este valor es la longitud del ciclo de  $n$ . En el ejemplo anterior, la longitud del ciclo del 22 es 16.

Para dos números cualquiera  $i$  y  $j$ , ud. tiene que determinar la longitud de ciclo máxima para todos los números entre  $i$  y  $j$ .

### The Input

The input will consist of a series of pairs of integers  $i$  and  $j$ , one pair of integers per line. All integers will be less than 1,000,000 and greater than 0.

You should process all pairs of integers and for each pair determine the maximum cycle length over all integers between and including  $i$  and  $j$ .

You can assume that no operation overflows a 32-bit integer.

### Entrada

La entrada consistirá de una serie de pares de enteros  $i$  y  $j$ , un par por línea. Todos los enteros serán menores que 1,000,000 y mayores que 0.

Ud. debe procesar todos los pares de enteros y para cada par determinar la longitud de ciclo máxima para todos los enteros entre -e incluyendo-  $i$  y  $j$ .

Puede asumir que ninguna operación producirá un overflow sobre enteros de 32-bits.

### The Output:

For each pair of input integers  $i$  and  $j$  you should output  $i$ ,  $j$ , and the maximum cycle length for integers between and including  $i$  and  $j$ . These three numbers should be separated by at least one space with all three numbers on one line and with one line of output for each line of input. The integers  $i$  and  $j$  must appear in the output in the same order in which they appeared in the input and should be followed by the maximum cycle length (on the same line).

### Salida:

Para cada par de enteros  $i$  y  $j$  de la entrada, ud. debe imprimir  $i$ ,  $j$ , y la longitud de ciclo máximo para los enteros entre e incluyendo  $i$  y  $j$ . Estos tres números deben estar separados por al menos un espacio en blanco y los tres deben estar en la misma línea, con una línea de salida para cada línea de entrada. Los enteros  $i$  y  $j$  deben aparecer en la salida en el mismo orden en el cual ellos están en la entrada y deben estar seguidos por la longitud de ciclo máxima (en la misma línea).

### Sample Input:

1 10  
100 200  
201 210  
900 1000

**Sample Output:**

1 10 20  
100 200 125  
201 210 89  
900 1000 174

**Ejemplo de entrada:**

1 10  
100 200  
201 210  
900 1000

**Ejemplo de salida:**

1 10 20  
100 200 125  
201 210 89  
900 1000 174