

O Problema $3n+1$

O problema (original em inglês) localiza-se [aqui](#).

Os problemas na Ciência da Computação são frequentemente classificados como pertencendo a determinadas classes. Nesse problema você vai analisar a propriedade de um algoritmo cuja classificação não é conhecida por todas as entradas possíveis.

Considere o seguinte algoritmo:

```
input n
print n
if  $n = 1$  then STOP
    if  $n$  is odd then  $n \leftarrow 3n+1$ 
    else  $n \leftarrow n/2$ 
GOTO 2
```

Dado 22 como entrada, a saída será a seguinte sequencia de números:

22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Considera-se que o algoritmo acima irá terminar (1 é escrito) para qualquer entrada com valor inteiro. Apesar da simplicidade do algoritmo, não se sabe se esse pressuposto é verdadeiro. Contudo, isso foi verificado para todos os inteiros n , tal qual $0 < n < 1.000.000$ (e para mais muitos valores adiante).

Dada uma entrada n , é possível determinar o número de valores escritos, incluindo o valor 1. Assim, para determinado n , chama-se **tamanho do ciclo** de n . No exemplo acima, o tamanho do ciclo de 22 é 16.

Para quaisquer dois números i e j , você deve determinar o tamanho máximo do ciclo de todos os números entre i e j , ambos inclusos.

Entrada

A entrada consiste numa série de pares de números i, j (um por linha). Todos os inteiros devem ser menores que 10.000 e maiores que 0.

Você deverá processar todos os pares de inteiros, onde, para cada par, deve-se determinar o tamanho máximo do ciclo perante todos os inteiros entre i e j , inclusos.

Você pode assumir que nenhuma das operações irá criar um overflow no inteiro de 32 bits.

Saída

Para cada entrada de pares inteiros i, j você deve dar como saída i, j e o tamanho máximo do ciclo para os inteiros de i até j . Esse três números devem ser separados por pelo menos um espaço em branco e estarem em linha individual (uma linha de entrada por uma de saída). Os inteiros i e j devem aparecer na saída na mesma ordem na qual eles aparecem na entrada, seguidos, logo em seguida, pelo tamanho máximo do ciclo (mesma linha).

Exemplo de entrada

```
1 10
100 200
201 210
900 1000
```

Exemplo de saída

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