

100 - O problema do $3n + 1$

Conhecimento Geral

Problemas em Ciência da Computação são frequentemente classificados como pertencentes a certas classes de problemas (por exemplo, NP, Insolúveis, Recursivos). Neste problema você estará analisando uma propriedade de um algoritmo cuja classificação não é conhecida para todas as entradas possíveis.

O Problema

Considere o seguinte algoritmo:

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

Dada a entrada 22, a seguinte sequência de números será impressa:
22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Há uma conjectura de que o algoritmo acima irá terminar (ou seja, o número 1 será impresso) para qualquer entrada inteira positiva. Desconsiderando a simplicidade do algoritmo, não é sabido se a conjectura é verdadeira. Ela foi verificada, no entanto, para todos os inteiros n tal que $0 < n < 1.000.000$ (e, de fato, para muito mais números maiores que isto).

Dada uma entrada n , é possível determinar a quantidade de números impressa (incluindo o 1). Para um dado n , isto é chamado tamanho cíclico de n . No exemplo anterior, o tamanho cíclico de 22 é 16.

Para dois números i e j , determine o tamanho cíclico máximo de todos os números de i até j .

A Entrada

A entrada consistirá de uma série de pares de inteiros i e j , um par de inteiros por linha. Todos os inteiros serão menores que 1.000.000 e maiores que 0.

Você processará todos os pares de inteiros e , para cada par, determinará o tamanho cíclico máximo de todos os inteiros de i até j .

Você pode pressupor que não haverá operações sobre inteiros de mais de 32 bits.

A Saída

Para cada par de inteiros de entrada i e j , você deverá imprimir como saída “ $i j max$ ” onde max é o tamanho cíclico máximo dos números de i até j . Estes três números deverão ser impressos na mesma linha e separados por ao menos um espaço em branco. Cada resultado (os três números) deverá aparecer em uma linha da saída. Os inteiros i e j deverão aparecer na mesma ordem em que eles apareceram na entrada e devem ser seguidos pelo valor do tamanho cíclico máximo (na 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
```