UVa 371

Contents

371 - Ackermann Functions

- http://uva.onlinejudge.org/index.php?option=com_onlinejudge&Itemid=8&category=5&page=show_problem&problem=307 - UVA 371

Summary

An Ackermann function is a function where you cannot determine the length of recursion based on the input value. In this problem, perform the instructions for one ackermann function : If **X** is even, divide by two, and if odd, Muliply by 3 and add 1.

Follow the instructions for all the numbers in each given range of two numbers, keeping record of the largest *cycle*. Print out the largest cycle size and the initial number.

Explanation

A "follow the instructions" problem. Bruteforce will work.

Optimizations

- There's no harm in memoizing solutions for individual numbers.

Gotchas

- As with UVa 100, the first number may be greater than the second.
- However, the output states that the lower and upper bounds must be printed in a certain order.

Input

```
  1 20
 35 55
2000 3000
```

```
3000 2000
   0 0
```

Output

```
Between 1 and 20, 18 generates the longest sequence of 20 values.
Between 35 and 55, 54 generates the longest sequence of 112 values.
Between 2000 and 3000, 2919 generates the longest sequence of 216 values.
Between 2000 and 3000, 2919 generates the longest sequence of 216 values.
```

Solutions

- UVa 371. Ackermann Function with JAVA (http://adilakhter.wordpress.com/2013/03/14/uva-371-ackermann-function/)

See Also

- Similar problems: UVa 100 UVa 694
- A two-parameter Ackermann function: NK1264 - http://acm.nankai.edu.cn/problem.php?problem=1264
- Wikipedia:Ackermann function
- Explanation on Collatz Problem a.k.a 3n+1 Problem (http://adilakhter.wordpress.com/2013/02/20/collatz-problem-a-k-a-3n1-problem/)