

Computer Science 2310

Homework 3

DUE: Tuesday November 10 at the start of class

Goals

- Practice writing a program that involves function usage and determination of extrema values.
- Practice incremental program development.

Assignment

Write a Java program complete with full documentation for the following specification.

Hailstone Program

The **hailstone** sequence, sometimes called the $3n + 1$ sequence, is defined by a function $f(n)$:

$$f(n) = \begin{cases} n / 2 & \text{if } n \text{ is even} \\ 3 * n + 1 & \text{otherwise, if } n \text{ is odd} \end{cases}$$

We can use the value computed by f as the argument of a new invocation of f as shown below; the successive values of n form the hailstone sequence.

```
while (n != 1)
{
    n = f(n);
}
```

Although it is conjectured that this loop always terminates, no one has been able to prove it. However, it has been verified by computer for an enormous range of numbers. Several sequences are shown below with the initial value of n on the left.

7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

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

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

14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

8 4 2 1

9 28 14 7 22 11 34 17 52 26 13 40 20 10 5 16 8 4 2 1

Write a program to find the values of n that yields the longest sequence. Prompt the user for two numbers, and limit the search for n to all values between the two numbers (including the endpoints).

Sample interactions your program should produce are given below. User's input is underlined.

Enter lower limit: 7
Enter upper limit: 20

18 produces the longest hailstone sequence between 7 and 20 inclusive.
It's length is 21.

Technical Issues

- The *hailstone sequence* is the sequence of numbers produced by the f function. The sequence terminates when the result of the function call is 1 .
- Step 1 in your program development should be to write a program that reads in a single number and then prints out the hailstone sequence for that number and also prints out the length of the sequence. The pseudocode for generating the sequence was previously mentioned. You will need to augment that logic to do the actual $f(n)$ calculation and print the numbers as well as prompt for the initial value.
- Besides your *main* method, you will implement the functions f and *lengthHail*. The template for the program is provided on the course WWW page.
- Step 2 in your program development is to write a version of your program that tests f and *lengthHail* to see if they work correctly for 1 value. Use one of the sample sequences specified on the previous page. Note that in the final version of the program, *lengthHail* simply returns the length of the sequence; it does not display any output on the screen. However, in order to check and see that it is working correctly, you will have to insert an appropriate output statement to ensure it is counting the correct sequence of numbers. Once you've checked the correctness, it is acceptable to simply comment out the output statement rather than delete it, in case further debugging is later necessary.
- Step 3 is to modify the program to complete the assignment by acquiring the numerical limits and calculating the value that produces the longest sequence.
- **NOTE:** programs that do not properly define the functions f and *lengthHail* will be severely penalized, even if the program produces the correct answer.
- Write down the results of at least 3 test cases other than the ones given above. Put your answers in the header comments at the top of your program. **WARNING:** do not use values bigger than 113382 in your test cases.
- This problem is worth **75 points**.
- **Submission information:** Name your file HailStone.java and place in a separate subdirectory. Use the command `submit h3 HailStone.java` to electronically submit your assignment.