A1-Do Not Fib

Stuart Thiel

May 19, 2023

Introduction

This assignment is worth **5%** of your grade. This is an **individual assignment** (so individual, I put *your* student ID on it) and you should not share your assignment with anyone else. The **assignment is due May 26th at 23:59, Montreal time**.

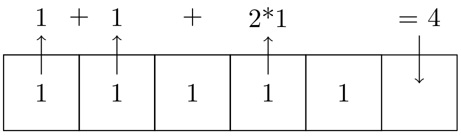
All submissions must go through Moodle. I like EAS, but the new VPN rules suck.

Q1) Do Not Fib!

Your task is to create a program named `FibonacciSequenceInspector.java`. This program should define a single class named FibonacciSequenceInspector, which should be part of the comp352.datastructures package. The main method of the FibonacciSequenceInspector class should accept a single argument, representing the desired length of the sequence to be generated.

The sequence generated should start with 5 number of '1s'. Each subsequent value in the sequence should be the sum of all previous values except for the ones 3 and 5 steps back in the sequence. Moreover, the value 4 steps back in the sequence should be multiplied by 2.

For example, after the initial sequence of '1s', this is how you would calculate the next value:



The generated sequence should be printed on a single line. In addition, the program should also output the number of times something is skipped.. It is crucial that the output format aligns with the example provided and the format that we're testing for with DebugRunner. Failing to adhere to this format will result in a score of zero.

1.1 Evaluation

In the zip file containing this assignment, you'll find additional files, notably `DebugRunner.class` and `config.xml`. The zip also includes several test inputs. Once you have a `.class` file, compiled from your written Java source, you can run `java DebugRunner` from the same folder to receive feedback. This requires command-line operation. Please seek assistance from your tutors if you're unsure how to proceed. Most assignments will be accompanied by this setup. The config.xml file will grade your assignment partially so you can gauge your progress. However, remember that the final marking will be based on a more comprehensive `config.xml`. Hence, it's crucial to follow assignment instructions meticulously.

1.2 Theory Questions

Answer your theory questions in comments at the top of your Java code. Include both questions and your answers.

Structured Insertion Sort Example

You might type: *java 8*

The output would look something like:

```
1 1 1 1 1 4 4 10
Calls: 20
```

Q2) Theory: Qualify This Recursion

In your own words, discuss the trade-offs between using recursion and using an iterative approach for this particular assignment. Limit your answer to 150 words.

Q3) Theory: Impact of Size

What sort of issues could potentially arise when working with very large sequences in your program? How could these issues be mitigated? Limit your answer to 150 words.