

Gavin Swiger

Documentation for Fib_Seq.java

Assignment Description:

Implement the Fibonacci function in both a recursive and iterative fashion. What's the runtime efficiency of each?

You can look up sample programs - there are many out there - but for your own experience, please type yours from scratch.

Turn in a chart of the results, with time on the Y axis, and input on the X axis, Please use nanosecond. **long** startTime = System.*nanoTime*(); This chart must not be hand written.

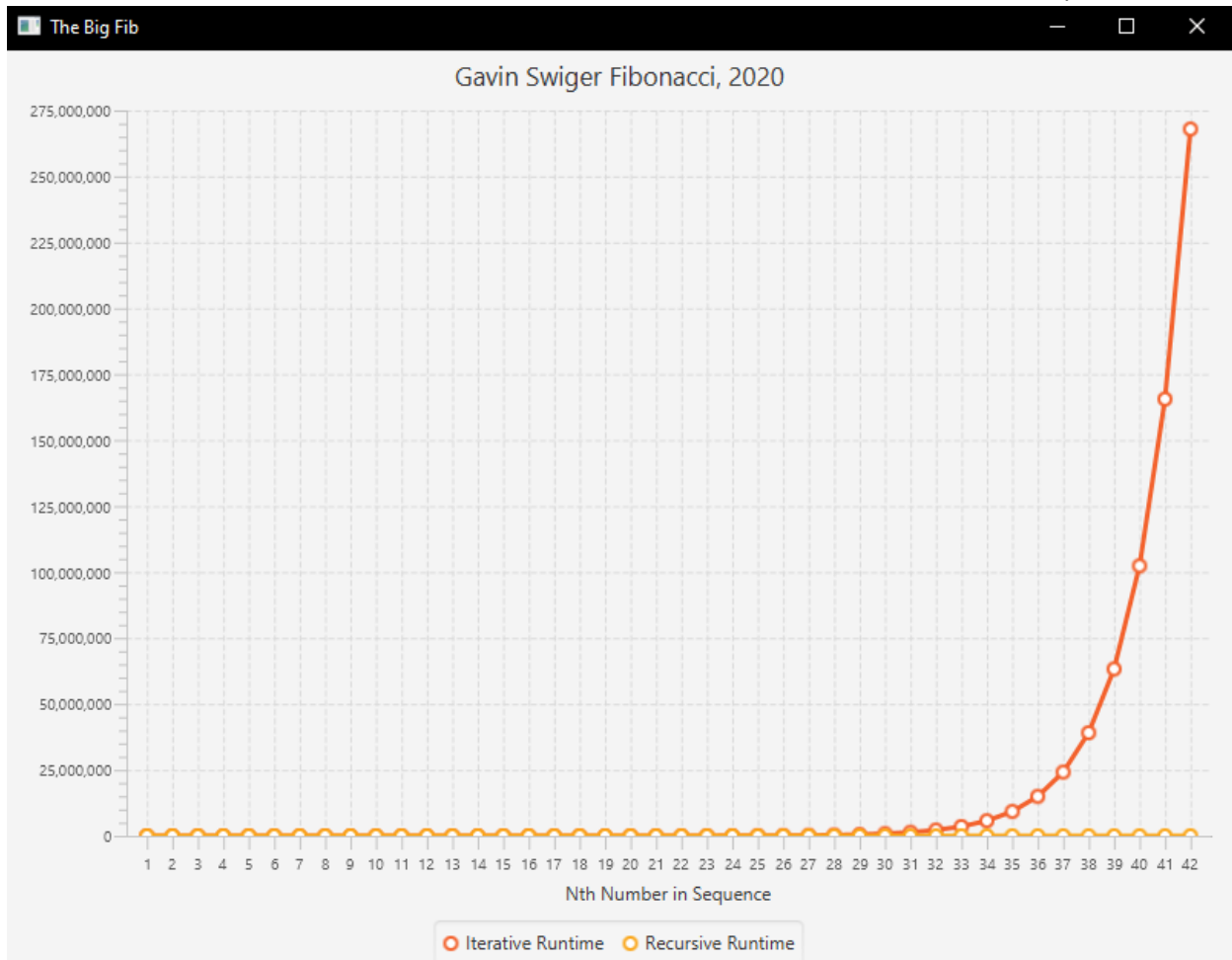
What it Does:

The program prompts user "Enter an integer for the Fibonacci Series:" in the console. The program then takes the user's integer and returns an integer that corresponds to that position in the Fibonacci series. The program also returns the run time for obtaining the corresponding number both recursively and iteratively.

Run Time efficiency:

The runtime of each function is kept by calling "`System.nanoTime()`" just before the function is called and just after the function returns the requested number from the Fibonacci series. The `startTime` is then subtracted from the `endTime` resulting in `totalRunTime`. This variable is divided by 20000 handed to the `getData` function to be added to the chart.

The performance difference of the two functions can be visualized using the below chart. The run time of the Iterative function is linear while the runtime of the recursive function is exponential.



Things of Note:

- The program uses ints variable types rather than long variable types so testing the run time the interactive function is limited.
- On my desktop going past the 50th number in the fibonnacci sequence (The user entering an int past 50) was very performance heavy and returns a number outside of the scope of an int variable anyways. (a number larger than 2,147,483,647).