Mod 5: Fibonacci

# Project

Our project for Module 5 is to implement a Fibonacci function - comparing both recursive and iterative versions.

# Tests

For this project I wrote my Fibonacci functions in Java. In the program I ran both recursive and iterative functions and used *System.nanoTime()* to record their run times in nanoseconds. I also had to make sure to initialize my first run and not record that data in order to get an accurate reading, otherwise the delay would give inaccurate results.

I’m using a chart to record their display each function and their times.

# Code

# Outcome

The Fibonacci sequence is the sum of the two numbers before it. Going from one number (5) and two numbers (5, 10) both functions are roughly comparable, but as the number value increased so did the runtime for our recursive function. Because we’re not storing any data, our function must calculate the sum of two numbers all down the line in our count. The recursive function gets increasingly slower as more function calls are needed. However, we only need to calculate the sum of two numbers. By saving only the last two sequence numbers, we can perform a loop to find additional Fibonacci sequences while never having to recalculate our previous positions.