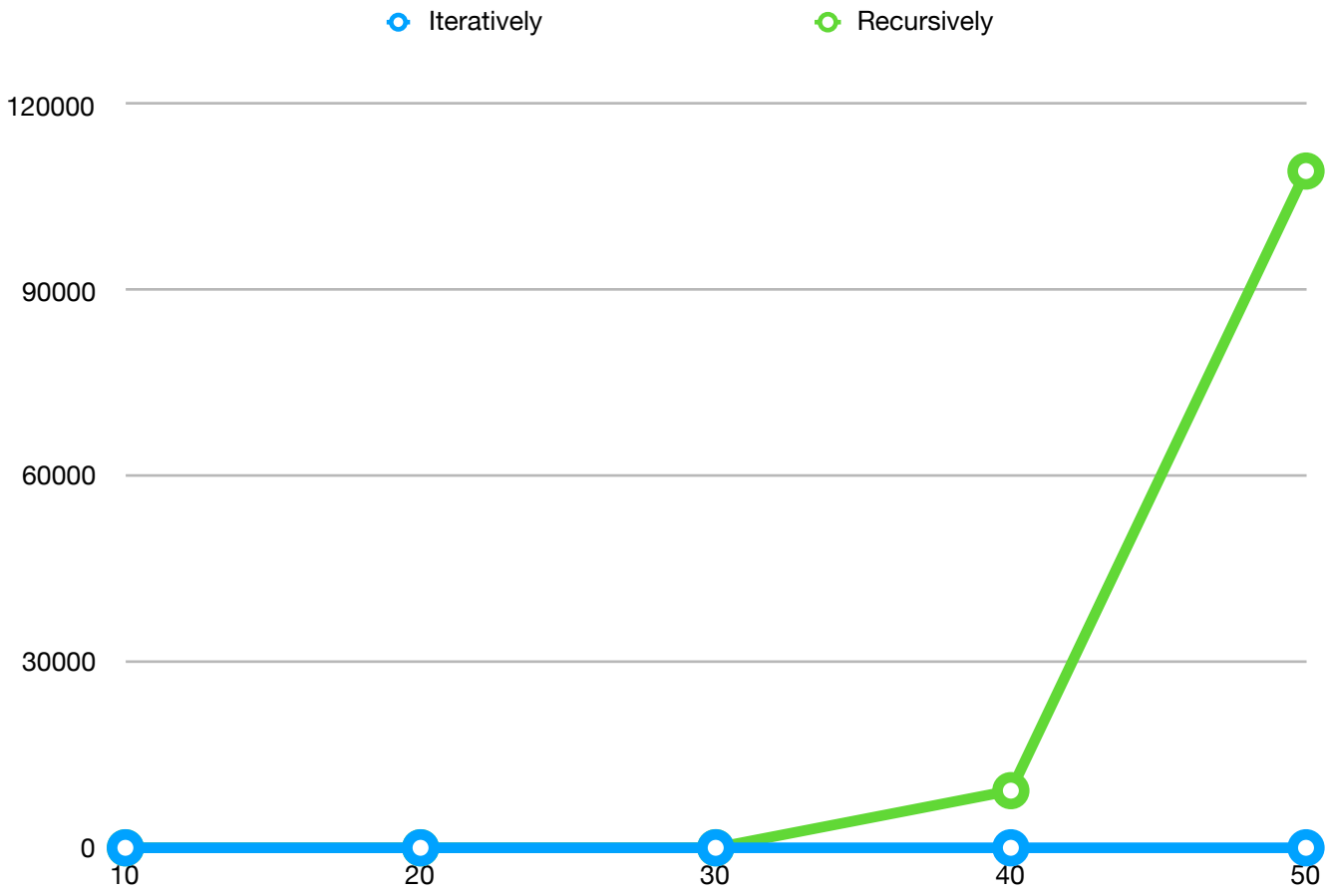
3-Screenshot testing verifying that your program actually generates a real Fibonacci series (in a PDF)

```
➜  COIS 2240 javac Fibonacci.java
➜  COIS 2240 java Fibonacci
Input the n for fibonacci: 15
----- n=15 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
Execution Time: 1ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377]
Execution Time: 0ms
➜  COIS 2240
```

4-Screenshot testing verifying that you ran the five different tests with different n values as instructed above (for both the iterative generation and recursive generation of the Fibonacci sequence, this should be in a PDF)

```
➜  COIS 2240 git:(master) ✗ javac Fibonacci.java
➜  COIS 2240 git:(master) ✗ java Fibonacci
----- n=10 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
Execution Time: 0ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34]
Execution Time: 1ms
----- n=20 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181]
Execution Time: 0ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181]
Execution Time: 0ms
----- n=30 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229]
Execution Time: 0ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229]
Execution Time: 8ms
----- n=40 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986]
Execution Time: 0ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986]
Execution Time: 845ms
----- n=50 -----
Iteratively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 16558
0141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049]
Execution Time: 0ms
Recursively Fibonacci: [0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610, 987, 1597, 2584, 4181, 6765, 10946, 17711, 28657, 46368, 750
25, 121393, 196418, 317811, 514229, 832040, 1346269, 2178309, 3524578, 5702887, 9227465, 14930352, 24157817, 39088169, 63245986, 102334155, 16558
0141, 267914296, 433494437, 701408733, 1134903170, 1836311903, 2971215073, 4807526976, 7778742049]
Execution Time: 109266ms
➜  COIS 2240 git:(master) ✗
```

5-Two graphs one for the iterative method and another for the recursive method that plots the times in relation to n (the size of the Fibonacci sequence being generated). Those two graphs should be side by side in a PDF. You may use excel or any other software to generate the graphs.



6-The analysis and comparison of the graphs should be in the same PDF as the actual graphs, the analysis should be brief.

The horizontal axis is the user-defined number n for fibonacci and the vertical axis is the execution time in milliseconds. We can see that when number n is less than 30, there is no obvious difference between the two functions. But when the number n reaches 40 and 50, the recursive function will need much more time than iterative function. That's because previous calculation results in recursive function will not be saved and used. While in iterative function, the previous results are saved in an array so they can be used for calculating the next fibonacci numbers. This will save a lot of time.

7-Screenshot of commit logs in a PDF.