Stephen Smith
CMSI 282
Dr. Dorin
2/4/2014

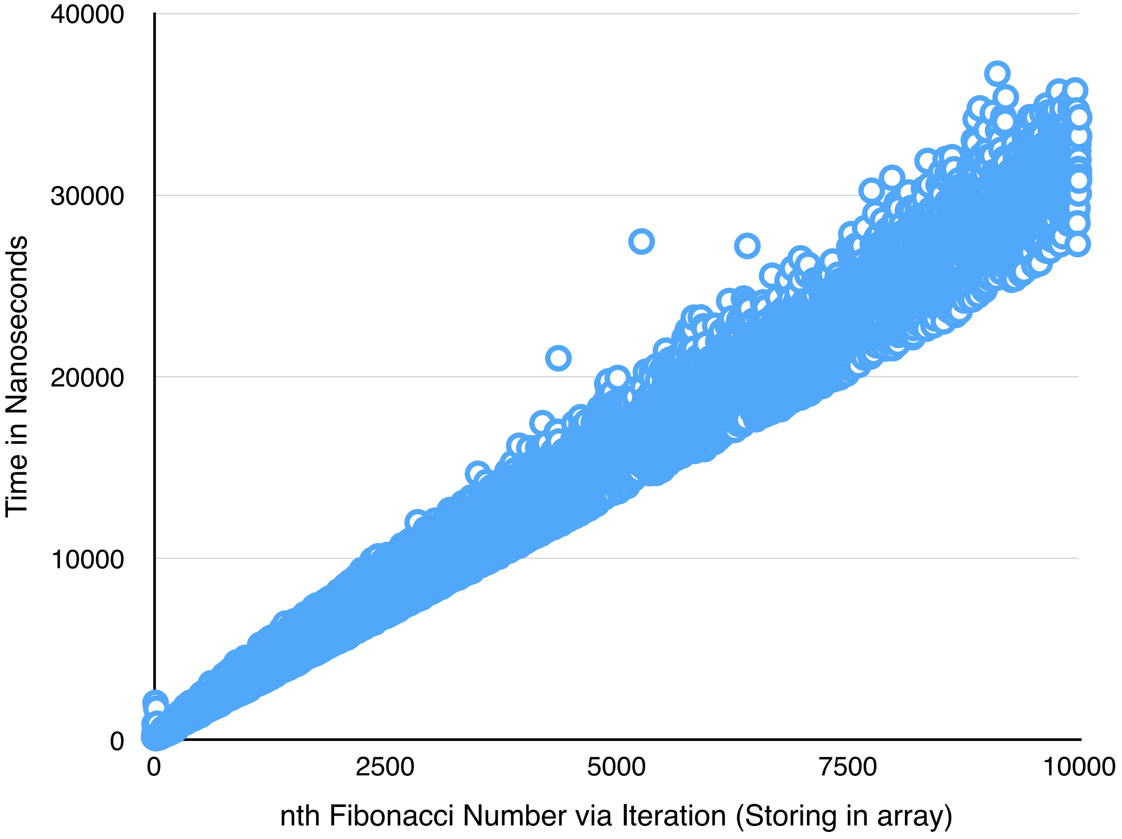
Fibonacci Function Evaluation

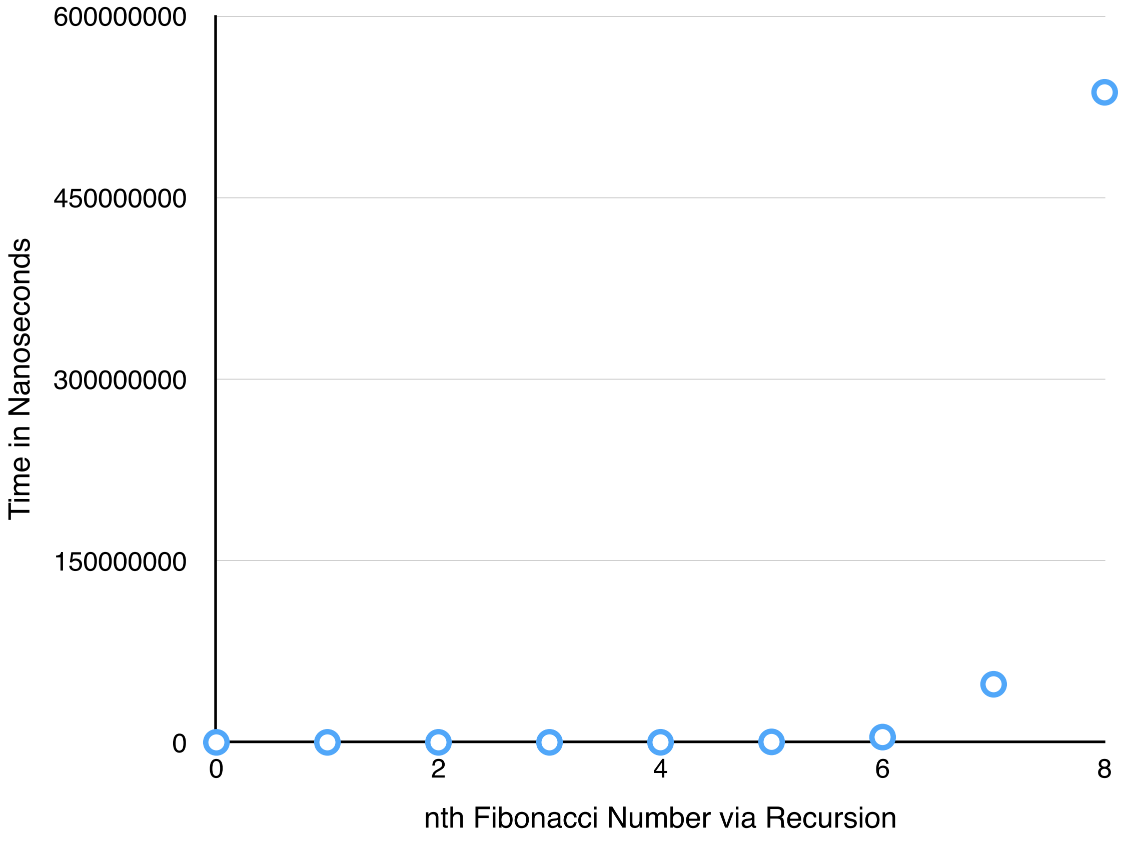
Problem Statement:
Compare the runtimes of recursive and iterative functions that generate the nth Fibonacci number.

Code:

```
import java.math.BigInteger;

public class fib {
    public static void main (String[]args) {

        int[] fibNumbers = {1, 5, 10, 15, 20, 25, 30, 35, 40};

        for (int j = 0; j < fibNumbers.length; j++) {


            long totalTime = 0;
            for (int i = 1; i <= 1000; i++) {
                long startTime = System.nanoTime();
                long fibNum = recursiveFib(fibNumbers[j]);
                long finalTime = System.nanoTime() - startTime;
                totalTime += finalTime;
            }
            long averageTime = totalTime / 1000;

            System.out.println(j + ", " + averageTime);
        }



        for (int i = 1; i <= 10000; i++) {
            long totalTime = 0;

            for (int j = 1;  j<= 1000; j++) {
                long startTime = System.nanoTime();
                long fibNumber = iterationFib(i);
                long finalTime = System.nanoTime() - startTime;
                totalTime = totalTime + finalTime;
            }

            long averageTime = totalTime / 1000;
            System.out.println(i + ", " + averageTime);
        }
    }
    public static long recursiveFib (int k) {
        switch (k){
        case 0: return 0;
        case 1: return 1;
        default: return recursiveFib(k - 2) + recursiveFib(k - 1);
        }
    }

    // better way, simple dynamic program
    public static long iterationFib (int k) {
        long[] solutions = new long[1+k];
        solutions[0] = 0;
        solutions[1] = 1;

        for (int i = 2; i <= k; i++) {
            solutions[i] = solutions[i-2] + solutions[i-1];
        }
        return solutions[k];
    }
}
```

Data:


40000
30000
20000
10000
0
Time in Nanoseconds
0
2500
5000
7500
10000
nth Fibonacci Number via Iteration (Storing in array)


600000000
450000000
300000000
150000000
0
Time in Nanoseconds
0
2
4
6
8
nth Fibonacci Number via Recursion