Homework 05 - fib of n

Focus: hash tables

Due: Mon Feb 10, 2020 (bring to class)

Reading:

- Sedgewick Java 2.3 Recursion, introcs.cs.princeton.edu/java/23recursion
- Prof Bill Recursion notes, wtkrieger.faculty.noctrl.edu/csc210-spring2020/docs/recursion_notes.pdf

thanks...yow, bill

Fibonacci coding

Step 1: Let's write some code to calculate the Fibonacci number, F(n):

 $F_n = F_{n-1} + F_{n-2}$ for $n \ge 2$, with $F_0 = 0$ and $F_1 = 1$

Write two methods to do this: 1) recursive, and 2) iterative. My simple solution will probably look something like this:

```
public class Homework05 {
  public static void main( String[] args) {
    // 1) say hello, then ask user for an int
    // 2) call fibRecursive and print the answer
    // 3) call fibIterative and print the answer
  }
  public static long fibRecursive( int n) {
    // code here, use recursion to calc Fib(n) and return
  }
  public static long fibIterative( int n) {
    // code here, use a loop to calc Fib(n) and return
  }
}
```

/* Hint: There's some helpful pseudocode in the Sedgewick section referenced above. */

Once you're running, what is Fib(10)? Fib(20)? Fib(40)? What is Fib(100)? (yikes!)

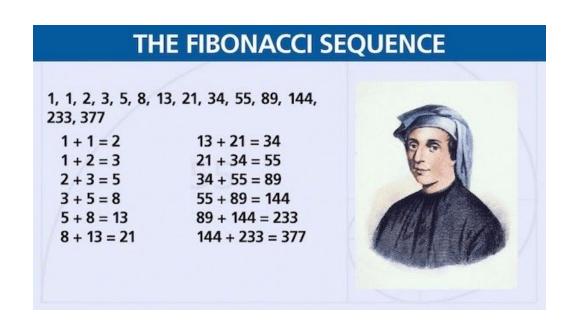
Step 2: Let's add some code to time how long our two methods take to run. I prefer the method System.currentTimeMillis() for benchmarking. It goes something like this:

```
long start = System.currentTimeMillis();
// call method to time
long end = System.currentTimeMillis();
long durationMillis = end - start;
```

Google for help. Also, if you find a better, funner way, then use it. And then you can show the rest of us.

Bonus: If the numbers in your Fib solution get too big for your program to handle, then google "java BigInteger". This is a class that can represent and do simple math operations on REALLY BIG integers.

This bonus step is *optional*, but BigInteger is really cool. Give it a try if you have time.



PS - Fibonacci is an Italian mathematician, born in 1170 (gasp!), en.wikipedia.org/wiki/Fibonacci