

AP COMPUTER SCIENCE A – FIBONACCI NUMBERS

The first nine Fibonacci numbers are

0 1 1 2 3 5 8 13 21.

Each Fibonacci number is obtained by adding the two previous ones.

The program `Fibonacci.java` calculates the n th Fibonacci number using a direct (or *iterative*) algorithm. The method `fibonacci` (lines 53 through 69) performs the calculation with a `for` loop on line 63 (and hence the term *iteration*, since it loops [or *iterates*] from 2 to $n-1$).

The program `RecursiveFibonacci.java` calculates the n th Fibonacci number using a *recursive* algorithm. The method `recursiveFibonacci` (lines 53 through 64) performs the calculation by calling method `recursion` (line 59). Note that in recursion, the method calls itself (line 72), *i.e.*, recursion calls recursion. When a method calls itself, this is called *recursive*.

Note that the largest Fibonacci number that can be represented by a `long` is the 93rd Fibonacci number, which is 7540113804746346429. This is why both programs (in method `readNth`) test if the input integer `nth` satisfies $0 < nth < 94$.

Class Work

1) Compile and run `Fibonacci.java`.

- a) What is the 50th Fibonacci number?
- b) The program also prints out the time, in milliseconds, that it took to calculate the number. What is this time?

2) Compile and run `RecursiveFibonacci.java`.

- a) What is the 50th Fibonacci number?
- b) The program also prints out the time, in milliseconds, that it took to calculate the number. What is this time?

3) Which program is better?