

L12 - Recursion Exercise

Part A – Fibonacci

A recursive algorithm can be very fast. Sometimes it's the only solution to a problem. Sometimes the recursive version of an algorithm has very poor performance compared to an iterative solution. In this part of the exercise, we'll empirically determine the performance of a recursive Fibonacci algorithm by measuring the elapsed time required to run the algorithm for a series of values. Fill in the missing code in the `fibonacci` method in the `FibonacciTest` class:

```
import java.util.*;

public class FibonacciTest {

    public static void main(String[] args) {
        Scanner myScanner = new Scanner(System.in);
        boolean done = false;
        while (!done) {
            System.out.print("Enter an integer or Q to quit: ");
            String answer = myScanner.nextLine();
            if (answer.equalsIgnoreCase("q")) done = true;
            else {
                int myLong = Integer.parseInt(answer);
                long startTime = System.currentTimeMillis();
                long fibN = fibonacci(myLong);
                long endTime = System.currentTimeMillis();
                double elapsedTime = (endTime - startTime)/1000.0;
                System.out.print("Fibonacci(" + myLong + ") = " + fibN);
                System.out.println(" took " + elapsedTime + " seconds");
            }
        }
        System.out.println("Goodbye!");
    }

    public static long fibonacci(long n) {
        if (n <= 2)
            //enter the code for the base case here
    }
}
```

```
else
    //enter the code for the recursive case here
}
```

Test your program using the following values and record your results:

fibonacci(10) = _____ and took _____ seconds
fibonacci(20) = _____ and took _____ seconds
fibonacci(30) = _____ and took _____ seconds
fibonacci(40) = _____ and took _____ seconds
fibonacci(50) = _____ and took _____ seconds

What do you notice about the time it takes to find the solution?

Part B – Computing a^n

Write a program that uses a recursive algorithm to compute the value of a^n where a is any number and n is an integer ≥ 0 . Prompt the user for values of a and n and display the result. Recall that a^n is the same as $aaa*a...*a$ (n times)

1. As you develop your algorithm, what is the base case?

What is the recursive case?

Part C – Reverse a String

Write a program that uses a recursive algorithm to print a string in reverse. For example, if the user entered "abcdefg" then the program would display "gfedcba". Some `String` class methods that might be useful are `length`, `substring`, and `charAt`. Prompt the user for a string and display the result.

1. What is the base case?
2. What is the recursive case?

Revision #1

Created 24 April 2025 22:14:19 by Brandon Duke

Updated 24 April 2025 22:14:26 by Brandon Duke