Assignment 10: Exercise: Fibonacci

Assignment Description:

Explore the Fibonacci sequence, a classic example of recursion, and implement a Java code snippet to calculate Fibonacci numbers.

**Questions**:

Fibonacci Sequence: Explain what the Fibonacci sequence is and how it is defined. Describe how recursion is naturally suited for calculating Fibonacci numbers.

**Solution** 1:

The Fibonacci sequence is a series of numbers where each number is the sum of the two preceding ones, usually starting with 0 and 1. Mathematically, it is defined as: F(0) = 0, F(1) = 1, and F(n) = F(n-1) + F(n-2) for n > 1. Recursion is naturally suited for calculating Fibonacci numbers because the problem can be divided into smaller, similar subproblems.

Question 2: Exercise: Fibonacci

Implement a Java recursive function to calculate the nth Fibonacci number. Explain the base cases and the recursive step in your solution.

**Solution** 2:

Here's a Java recursive function to calculate the nth Fibonacci number:

java

public int fibonacci(int n) {

if (n <= 1) { // Base case

return n;

} else {

// Recursive step: F(n) = F(n-1) + F(n-2)

return fibonacci(n - 1) + fibonacci(n - 2);

}

}

The base cases are when n is 0 or 1, and the recursive step calculates the Fibonacci number by summing the results of the previous two Fibonacci numbers.