Solution Review: Recursion

In this review, we give a detailed analysis of the solution to this problem.

We'll cover the following

Solution: Recursion

```
recursiveFibonacci <- function(testVariable)
{
    if(testVariable <= 1)
    {
        return(testVariable)
    } else
    {
        return(recursiveFibonacci(testVariable - 1) + recursiveFibonacci(testVariable - 2))
    }
}

# Driver Code
recursiveFibonacci(0)
recursiveFibonacci(1)
recursiveFibonacci(2)
recursiveFibonacci(5)
recursiveFibonacci(6)
```

Explanation

To find the element placed at the input index n in the Fibonacci series we need the elements placed at $(n-1)$ and $(n-2)$ positions.

For example, if current index = n then

```
fibonacciSeries[n] = fibonacciSeries[n-1] + fibonacciSeries[n-2].
```

This is a good situation where recursion can be used. Recursion means *function calling itself*, in the code above `recursiveFibonacci()` function calls itself with a lesser value several times. A termination condition is very important in recursion function: in this case, it is

```
if(testVariable <= 1)
{
   return(testVariable)
}
```

so that the code returns directly for the lowest indexes.

In the next lesson, we have a short quiz to test your concepts.