Solution Review: Calculate nth Fibonacci Number Using Recursion

Let's go over the solution review of the challenge given in the previous lesson.

We'll cover the following

Solution

Press the **RUN** button and see the output!

```
 7
 8    // Base Case
 9    if (n == 0) {
10      return 0;
11    }
12    else if (n == 1) {
13      return 1;
14    }
15
16    // Recursive Case
17    else {
18      return Fibonacci(n - 1) + Fibonacci(n - 2);
19    }
20
21  }
22
23  // main function
24  int main() {
```

```
25    // Initialize variable n
26    int n = 4;
27    // Declare variable result
28    int result;
29    // Call fibonacci function in main and store its output in result
30    result = Fibonacci(4);
31    // Print value of result
32    cout << n << "th Fibonacci number = " << result;
33    return 0;
34  }
```

```
Output                                                    1.83s

 4th Fibonacci number = 3
```

Explanation#

fibonacci function

The recursive `fibonacci` function takes a value of type `int` in its input parameters and returns the Fibonacci number at that value in the output.

Recursive case

Each element in fibonacci is a sum of its previous two elements. We recursively sum the last two elements until the base case. `fibonacci` returns the sum of `fibonacci (n-1) + fibonacci (n-2)`. This is the recursive case.

First base case

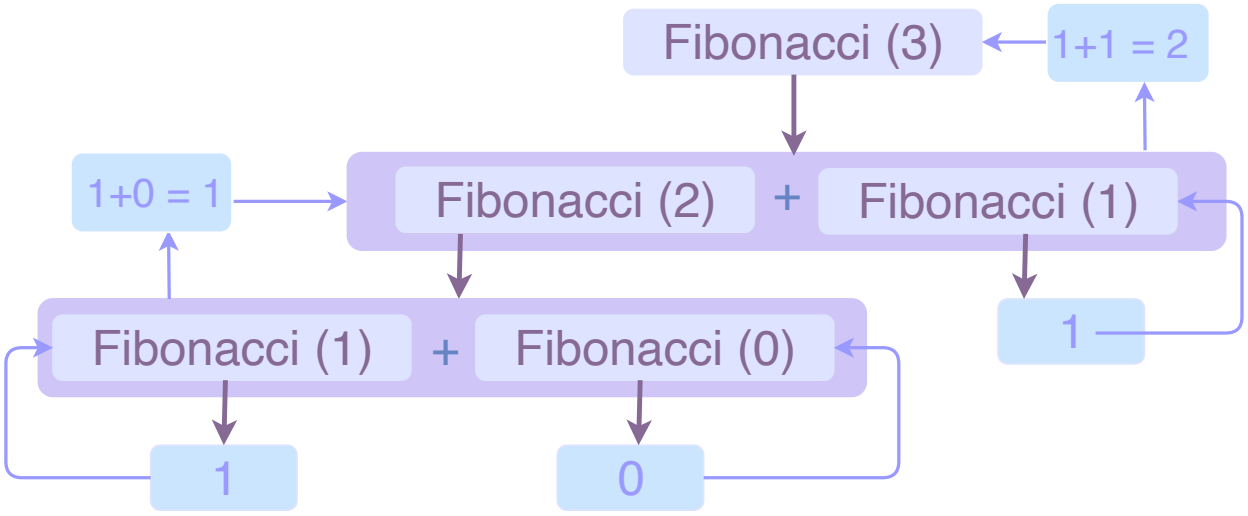
As the fibonacci 1st element is 1, if `n = 0`, the function terminates after returning **0** to the calling function.

Second base case

As the fibonacci 1st element is 1, if `n = 1`, the function terminates after returning **1** to the calling function.

Let's run our code for `n = 4` and see what happens inside the recursive `fibonacci()` function.

> In the Fibonacci function, there are two recursive calls in the function body. Therefore, it is known as a binary recursion.



In **Fibonacci(3)** will add the result of **Fibonacci(2)** and **Fibonacci(1)** and return it to the main function

10 of 10

Let's wrap up this chapter by completing a quiz in the upcoming lesson.