Fibonacci

anofiujelailah

September 2020

To properly explain why it is a bad idea to use recursion method to find the fibonacci of a number, let's analyze the following C++ code.

```cpp
#include <iostream>

using namespace std;
int fibonacciRecursion(int nthNumber) {
        //use recursion
        if (nthNumber == 0) {

            return 0;

        } else if (nthNumber == 1) {

            return 1;
        }
     return fibonacciRecursion(nthNumber - 1) + fibonacciRecursion(nthNumber - 2);
    }
    int main(){
    int n = 25;

    // Check n for Proth Number
    cout << (fibonacciRecursion(n));
}
```

Listing 1: Recursion example

This code uses the recursion algorithm to find the Fibonacci of a number. If the given number is equal to 0 and 1 we return both given numbers.

However, if the given number is greater than 0 and 1, we make two recursive calls where we add both calls with the nthNumber minus 1 and 2.

This will work perfectly when we pass integers' 1 to 5. However, higher numbers like 50, 100, and above will take so much longer.

The reason for this delay is the heavy usage of the stack memory in each recursive call.

A better approach to this would be iteration, which memorizes and stores each Fibonacci calculated.