Question 8.1 from CTCI: Fibonacci

October 18, 2017

Question

Write a method to generate the nth Fibonacci number.

Explanation and Algorithm

There are two solutions to this problem:

The first solution is an iterative solution. To generate the nth Fibonacci number, you add the $(n-1)$th and $(n-2)$th Fibonacci numbers. We can loop through the numbers from 3 to n and calculate each Fibonacci number in the sequence, storing the intermediate results which will then be used to calculate the next Fibonacci number. We loop through until we calculate the nth Fibonacci number.

For the second solution, instead of calculating each Fibonacci number up to the nth by looping iteratively, we can handle these sub-calculations through recursion. For the base case, we want our recursion to stop when we call the method on n = 0 and n = 1. If we were to continue recursion for these values, we would then be recursively calling the method with a n that is less than zero which is our error case. So we return 0 as the 0th Fibonacci number and 1 as the 1st Fibonacci number as our base case. Since we want to compute the nth Fibonacci number by recursively computing the $(n-1)$th and $(n-2)$th Fibonacci numbers, we can then return the following: $Fib(n-1) + Fib(n-2)$. The method will be called twice, once with $n-1$ as the argument and once with $n-2$ as the argument. The returned values will be added together to give the nth Fibonacci number, which is returned by the method.

Hints

1. Recall how we calculate the Fibonacci Sequence: $Fib(n-1) + Fib(n-2)$. How might you use this to calculate the nth Fibonacci number?

2. Consider a recursive solution. What would be your base case? When would you want to stop recursively calculating a Fibonacci number?

3. Consider an iterative approach instead. Is there a way to keep track of each intermediate Fibonacci number until you calculate the nth number?

Code

```
/*Answer 1 */
public static int Fib(int n){
   if(n < 0){
      return -1;
   }

   if(n == 0){
      return 0;
   }

   int a = 1;
   int b = 1;

   for(int i = 3; i <= n; i++){
      int c = a + b;
      a = b;
      b = c;
   }

   return b;
}
```

```
/* Answer 2 */
public static int Fib(int n){
   if(n < 0){
      return -1;
   }

   if(n == 0){
      return 0;
   } else if(n == 1){
      return 1;
   }

   return Fib(n -1) + Fib(n - 2);
}
```

Run time analysis

1. Iterative Solution: $O(n)$.
2. Recursive Solution: $O(2^n)$.

Sources

Question, answer and other material taken from Cracking the Coding Interview 4th edition by Gayle Laakmann McDowell.