

Exercise on recursion - 44251017

HUANG,Jiahui

Show an efficient algorithm for the following extended version of Fibonacci numbers:

$F_0 = F_1 = 0, F_2 = 1$

$F_{n+3} = F_n + F_{n+1} + F_{n+2} \ (n \geq 0)$

Show the time complexity of your algorithm.

The time complexity of this algorithm is $O(n)$.

```
1  public class Recursion {
2      public static int getFibonacci(int n) {
3          if (n == 0 || n == 1) {
4              return 0;
5          }
6          if (n == 2) {
7              return 1;
8          }
9
10         int a = 0, b = 0, c = 1; // F0, F1, F2
11         for (int i = 3; i <= n; i++) {
12             int ret = a + b + c;
13             a = b;
14             b = c;
15             c = ret;
16         }
17         return c;
18     }
19
20     public static void main(String[] args) {
21         Scanner scanner = new Scanner(System.in);
22         int n = scanner.nextInt();
23         System.out.println("F" + n + " = " + getFibonacci(n));
24     }
25 }
26
27 // Time Complexity:
28 // Each loop's time complexity is O(1)
29 // loop n times.
30 // Therefore, the total time complexity is O(n).
```