

## 1 Questão Única

Considere a sequência de Fibonacci.

1. Escreva um algoritmo recursivo simples para imprimir a sequência de Fibonacci, até um índice dado, e o número de chamadas recursivas executadas.
2. Melhore a sua implementação usando técnicas de programação dinâmica, por exemplo, salvando os valores intermediários em um array.
3. Escreva um algoritmo para gerar o n-ésimo número de Fibonacci usando um algoritmo que faça apenas n chamada recursivas.

Type an index of the Fibonacci sequence: 20

N	Fib(N)	Dumb	Smart
0	0	1	0
1	1	1	1
2	1	3	2
3	2	5	3
4	3	9	4
5	5	15	5
6	8	25	6
7	13	41	7
8	21	67	8
9	34	109	9
10	55	177	10
11	89	287	11
12	144	465	12
13	233	753	13
14	377	1219	14
15	610	1973	15
16	987	3193	16
17	1597	5167	17
18	2584	8361	18
19	4181	13529	19
20	6765	21891	20

[1, 1, 2, 3, 5, 8, 13, 21, 34, 55, 89, 144, 233, 377, 610]  
The 15th Fibonacci number is: 610

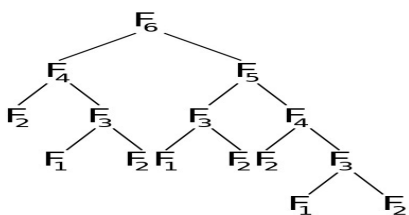
Algumas sugestões e requerimentos da implementação:

- Imprima uma tabela conforme explicitado acima, indicando o valor de  $N$ , da função  $FIB(N)$ , e o número de chamadas recursivas dos métodos simples e do esperto.
- O número de chamadas recursivas do algoritmo simples é  $O((\frac{1+\sqrt{5}}{2})^n) = O(1.618^n)$ , ou seja, para  $N = 20$  há 21891 chamadas.
- Nestes locais há uma boa discussão de algoritmos para geração da sequência de Fibonacci:  
[http://en.wikipedia.org/wiki/Fibonacci\\_number](http://en.wikipedia.org/wiki/Fibonacci_number)  
<http://functionspace.com/articles/32/Fibonacci-series-and-Dynamic-programming>  
<http://algorithms.tutorialhorizon.com/introduction-to-dynamic-programming-fibonacci-series>
- Escolha os componentes da interface mais adequados à entrada do índice da sequência (valor de  $N$ ) e para a exibição da tabela.

## 1. Solução

O problema da solução recursiva **ingênua** da sequência de Fibonacci, é a grande quantidade de chamadas recursivas repetidas executadas.

```
function fiboDumb ( n: longint ): longint;
begin
    if ( n <= 1 ) then
        fiboDumb := n
    else
        fiboDumb := fiboDumb(n-2)+fiboDumb(n-1);
    end;
end;
```



Na Ciência da Computação, programação dinâmica é uma técnica para solução de problemas complexos pela sua subdivisão em um conjunto de subproblemas mais simples, que são resolvidos apenas uma vez, e cujas soluções são armazenadas em alguma estrutura de dados em memória. Na próxima vez que o mesmo subproblema aparecer, ao invés de recomputar sua solução, basta consultar a solução computada previamente.

No caso da sequência de Fibonacci, é possível armazenar o resultado da  $n$ -ésima chamada recursiva, na posição  $n$  de um array, de modo que uma nova chamada recursiva só seja executada se não houver um valor armazenado na posição correspondente do array.

Quanto aos componentes da interface, é necessário apenas um TEdit para entrada do índice da sequência, e a exibição da tabela de resultados pode ser feita diretamente na console ou, para aqueles que desejarem estudar um pouco adiante, utilizando um TListBox.