SEQUÊNCIA DE FIBONACCI – ESFORÇO COMPUTACIONAL

PARTE I – Método iterativo

Os algoritmos apresentados a seguir são equivalentes e têm por objetivo determinar o n-ésimo termo da seqüência de Fibonacci (1, 1, 2, 3, 5, 8,...) definida pela relação de recorrência

$$F_k = F_{k-2} + F_{k-1}, \quad k = 3, 4, 5, \ldots$$

$$F_1 = 1, F_2 = 1.$$

1) O algoritmo Fibo descrito a seguir em pseudocódigo é recursivo e captura essa recorrência, fornecendo como saída o n-ésimo termo da sequência de Fibonacci:

Fibo(n)

```
se (n < 3)    então  x ← 1;
              senão  x ← Fibo(n-2) + Fibo(n-1);
Fibo ← x;
```

2) O algoritmo Fibonacci1 é equivalente ao algoritmo Fibo, mas não é recursivo. A variável j contém a soma dos dois termos anteriores da sequência.

Fibonacci1(n)

```
i ← 1;   j ← 0;
para  k  de 1 até n  repita  [ j ← i + j;   i ← j - i ] ;
Fibonacci1 ← j;
```

3) O algoritmo Fibonacci2 é equivalente aos dois algoritmos anteriores e também não é recursivo. A variável k contém a soma dos dois termos anteriores da sequência.

Fibonacci2(n)

```
j ← 1;   k ← 0;
para  p  de 1 até n repita [ i ← j;   j ←k;   k ← i + j ] ;
Fibonacci2 ← k;
```

4) O algoritmo Fib2 é equivalente aos algoritmos anteriores e chama o procedimento recursivo TentarSomar. Os parâmetros de entrada em TentarSomar são o valor de n (número de termos) e os dois valores iniciais da sequência. Esse algoritmo serve para determinar o n-ésimo termo de qualquer sequência que tenha a mesma recorrência que a sequência de Fibonacci.

Fib2(n)

```
f  ←  TentarSomar(n,1,1);
Fib2 ← f;
```

Algoritmo TentarSomar
Objetivo: Calcular o n-ésimo termo de uma seqüência de Fibonacci, dados os dois primeiros termos.
Parâmetros de entrada: n, t1,t2 (inteiro)
Parâmetros de saída: s (inteiro)
Pré-condição: n>0

TentarSomar(n,t1,t2)

```
se  (n=1) então  s ← t1
          senão  se (n=2) então  s ← t2
                         senão  [ c ← t1 + t2;
                                  s ← TentarSomar(n-1, t2, c) ] ;
TentarSomar ← s;
```

PARTE II – Método "direto"

5) O algoritmo Fibonacci descrito a seguir em pseudocódigo traduz um método direto do cálculo do n-ésimo termo da sequência de Fibonacci, por meio da sua fórmula de termo geral:

$$f(n) = \frac{1}{\sqrt{5}}\left(\left(\frac{1+\sqrt{5}}{2}\right)^n - \left(\frac{1-\sqrt{5}}{2}\right)^n\right), n = 1,2,\ldots$$

n	E(n)
1	0
2	0
3	3
4	6
5	12
6	
7	
8	
9	
10	
50	
100	

Fibonacci(n)

```
raiz ← sqrt(5);
inv ← 1/raiz;
p1 ← (1 + raiz)/2;
p2 ← (1 - raiz)/2;
termo ← inv x (pote(p1,n) - pote(p2,n));
Fibonacci ← termo
```

Esforço computacional do algoritmo recursivo Fibo

Vamos considerar somente a contagem das operações aritméticas para determinar o esforço computacional do algoritmo Fibo considerando algumas instâncias de entrada e a seguir a função de complexidade do algoritmo.

Fibo(n)

```
se (n < 3)    então   x ← 1;
              senão   x ← Fibo(n-2) + Fibo(n-1);
Fibo ← x;
```

Para n=1 ou n=2, nenhuma operação aritmética é executada.

Para n maior do que 2, são executadas 3 operações aritméticas e posteriormente a soma dos dois valores devolvidos por Fibo(n-2) e Fibo(n-1).

E(50) = 3 + E(48) + E(49) =

= 3 + 14.422.580.925 + 23.336.226.144 = 37.758.807.072

E(100) = 3 + E(98) + E(99) =

= 3 + 405.905.557.034.120.380.000 +656.768.987.503.665.740.000 = 106.267.454.453.778.610.000

Se considerarmos que o processador executa uma operação aritmética em 0.2 ns = $2x10^{-10}$ s, temos uma estimativa do tempo de execução da chamada de Fibo(100): aproximadamente 23 séculos.

EXERCÍCIOS – preparação para o laboratório

1) Faça um programa para calcular e mostrar os 10 primeiros termos da sequência de Fibonacci, usando um dos algoritmos dados.
   double Fibonacci(int)

2) Faça um programa para calcular e mostrar o esforço computacional na execução do algoritmo Fib2, considerando somente as operações aritméticas efetuadas no cálculo do 10º termo da sequência de Fibonacci.
   double Fibonacci(int)

3) Faça um programa para determinar o tempo de execução gasto no cálculo do 10º termo da sequência de Fibonacci, usando o algoritmo recursivo Fibo. Usar a função time para medição do tempo.
   double Fibonacci(int)