

Lista 8 - Algoritmos Recursivos

1. Escreva um algoritmo iterativo para avaliar $a * b$ usando a adição, onde a e b são inteiros não-negativos.

2. Escreva uma definição recursiva de $a + b$, onde a e b são inteiros não-negativos, em termos da função sucessora *succ* definida como:

```
int succ(int x) {
    return (++x);
}
```

3. Escreva uma versão iterativa do algoritmo de busca binária. (*Dica:* Modifique diretamente os valores de *low* e *high*.)

4. Conte o número de adições necessárias para computar $fib(n)$ para $0 \leq n \leq 10$, por meio dos métodos iterativo e recursivo. Existe algum tipo de padrão?

5. Determine o que a seguinte função recursiva em C calcula. Escreva uma função iterativa para atingir o mesmo objetivo.

```
int func(int n) {
    if (n == 0) {
        return 0;
    }
    return n + func(n-1);
} /* fim func */
```

6. A expressão em C, $m \% n$, resulta o resto da divisão de m ao dividir por n . Defina o máximo divisor comum (MDC) de dois inteiros, x e y , por:

$$mdc(x, y) = \begin{cases} y & \text{se } y \leq x \text{ E } x \% y = 0 \\ mdc(y, x) & \text{se } x < y \\ mdc(y, x \% y) & \text{caso contrário} \end{cases} \quad (1)$$

Escreva uma função recursiva em C para calcular $mdc(x, y)$. Descubra um método iterativo para calcular essa função.

7. Defina uma **sequência de Fibonacci generalizada**, de f_0 a f_1 como sequência $gfib(f_0, f_1, 0), gfib(f_0, f_1, 1), gfib(f_0, f_1, 2), \dots$, onde:

```
gfib(f0, f1, 0) = f0
gfib(f0, f1, 1) = f1
gfib(f0, f1, n) = gfib(f0, f1, n-1) + gfib(f0, f1, n-2) se n > 1
```

Escreva uma função recursiva em C para calcular $gfib(f_0, f_1, n)$. Descubra um método iterativo para calcular essa função.