

Algoritmos e Programação 2

Recursão

1. Implemente um programa capaz de realizar operações recursivas em uma lista de números naturais, contendo as seguintes opções:
 - (a) **Imprimir conteúdo:** imprime todos os elementos presentes na lista, recursivamente.
 - (b) **Imprimir menor elemento:** encontra o elemento com menor valor na lista, recursivamente, e o imprime.
 - (c) **Imprimir maior elemento:** encontra o elemento com maior valor na lista, recursivamente, e o imprime.
 - (d) **Imprimir soma dos elementos:** calcula a soma do valor de todos elementos da lista, recursivamente, e imprime.
 - (e) **Imprimir produto dos elementos:** calcula o produto do valor de todos elementos da lista, recursivamente, e imprime.
 - (f) **Buscar elemento:** se a lista estiver ordenada, deverá ser realizada a busca binária recursiva para encontrar o valor-alvo. O retorno desta função é o índice do elemento que contém o valor buscado. Caso contrário, realiza busca linear para encontrar o valor informado.
2. Critique a seguinte função recursiva (ela promete encontrar o valor de um elemento máximo de $v[0..n-1]$):

```
int maximo_rA (int n, int v[]) {  
    int x;  
    if (n == 1) return v[0];  
    if (n == 2) {  
        if (v[0] < v[1]) return v[1];  
        else return v[0];  
    }  
    x = maximo_rA (n-1, v);  
    if (x < v[n-1]) return v[n-1];  
    else return x;  
}
```

3. Exponenciação. Escreva uma função recursiva que receba inteiros positivos k e n e calcule k^n . (Suponha que k^n cabe em um int). Quantas multiplicações sua função executa aproximadamente?

4. Escreva uma função recursiva que calcule a função de Ackermann definida da seguinte maneira:

$$A(m, n) = \begin{cases} n + 1 & \text{se } m = 0 \\ A(m - 1, 1) & \text{se } n = 0 \\ A(m - 1, A(m, n - 1)) & \text{se } m > 0 \text{ e } n > 0 \end{cases}$$

5. Implemente o método recursivo para encontrar o n -ésimo termo da função de Fibonacci. A rotina deverá retornar o valor de $F(n)$ e a quantidade de vezes que a recursão foi chamada.
6. Que tipos de problemas um algoritmo recursivo é capaz de resolver?
7. Critique a documentação do seguinte código:

```
// Recebe um vetor de tamanho n
// e devolve a soma do vetor.
int soma (int v[], int n) {
    return sss (v, n+1);
}
int sss (int v[], int n) {
    if (n == 1) return 0;
    return sss (v, n-1) + v[n-1];
}
```

8. Seja a somatória abaixo. Faça uma função recursiva para realizar o cálculo.

$$\sum_{i=1}^n i^2$$