INSTITUTO FEDERAL
Catarinense
Campus Blumenau

Algoritmos

Bacharelado em Ciência da Computação
Prof. Paulo César Rodacki Gomes

Lista de exercícios - AER-Alg-30

Exercícios sobre recursividade em Python

O problemas desta lista devem ser resolvidos por meio de implementação de funções recursivas em Python. Caso você utilize funções, para cada um dos exercícios, crie um programa main para executar suas funções. **OBS.: utilize as mesmas regras das listas anteriores para dar nomes aos arquivos.**

1. **Fatorial**. Implementa uma função recursiva para calcular o fatorial de um número inteiro positivo. O fatorial é denotado pelo símbolo de exclamação "!" e é definido da seguinte forma: 1! = 1 e n! = n x (n-1)!, para n>1.

2. **Sequencia de Fibonacci.** A série de Fibonacci é uma sequencia de F_n números inteiros no qual um termo é definido pela soma dos dois termos anteriores. Os primeiros termos F_i da sequencia são 0, 1, 1, 2, 3, 5, 8, 13, etc. Portanto, o enésimo termo da sequencia é definido por $F_n = F_{n-1} + F_{n-2}$, sendo $F_0 = 0$ e $F_1 = 1$. Escreva uma função Python recursiva que recebe como parâmetro um valor inteiro n, e retorna o enésimo termo da sequencia de Fibonacci.

3. **Paíndromo**. Faça uma função Python **recursiva** que recebe uma string e retorne um valor lógico indicando se ela é ou não é um palíndromo. OBS: Um palindromo é uma palavra ou frase, que é igual quando lida da esquerda para a direita ou da direita para a esquerda (Espaços em branco e sinais de pontuação devem ser descartados). Exemplo de palindromo: "saudavel leva duas".

4. **Fibonacci com memorização de resultado**. Escreva uma nova versão da sua função recursiva do exercício 2 (Fibonacci) utilizando a técnica de memorização de resultado para melhorar desempenho e consumo de memória.

5. **Fatorial com memorização de resultado**. Escreva uma nova versão da sua função fatorial recursiva utilizando a técnica de memorização de resultado para melhorar desempenho e consumo de memória.

6. **Potenciação recursiva**. Implemente uma função recursiva em Python para calcular x^n. A função deve receber os dois parâmetros numéricos x e n, sendo n um numero inteiro não negativo. Obs.: lembre-se do caso base e do caso geral da potenciação para criar sua implementação recursiva.

7. **Soma recursiva**. Usando recursividade, implemente uma função Python que recebe uma lista de valores numéricos e retorne a soma de todos os valores da lista.