CENTRO FEDERAL DE EDUCAÇÃO TECNOLÓGICA DE MINAS GERAIS

CAMPUS TIMÓTEO

Lista de Exercícios 1 **Alg. e Estruturas de Dados I**

RECURSIVIDADE

EXERCÍCIO 1
Seja a série de Fibonacci:

1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

que pode ser definida recursivamente por:

$$Fibo(n) = \begin{cases} 1, & se\ n = 1\ e\ n = 2 \\ Fibo(n-1) + Fibo(n-2), & se\ n > 2 \end{cases}$$

Então escreva:
a) Uma função recursiva que gere o termo de ordem n da série de Fibonacci.
b) Um algoritmo que, utilizando a função definida acima gere a série de Fibonacci até o termo de ordem N.

EXERCÍCIO 2
O máximo divisor comum (MDC) de dois números inteiros x e y pode ser calculado usando-se uma definição recursiva:

$$MDC(x, y) = \begin{cases} MDC(x - y, y), & se\ x > y \\ MDC(y, x), & se\ x < y \\ x, & se\ x = y \end{cases}$$

Exemplo:
MDC (10 , 6) = MDC (4 , 6) = MDC (6 , 4) = MDC (2 , 4) = MDC (4 , 2) = MDC (2 , 2) = 2
Então, pede-se que seja criada uma função recursiva para descrever tal definição. Crie, também, um algoritmo que leia os dois valores inteiros e utilize a função criada para calcular o MDC de x e y, e imprima o valor computado.

EXERCÍCIO 3
Implemente uma função recursiva que receba por parâmetro dois valores inteiros x e y e calcule e retorne o resultado de x^y (x elevado a y) para o programa principal.

EXERCÍCIO 4
Implemente um procedimento recursivo para imprimir todos os números naturais de 0 até N (lido) em ordem "crescente".

EXERCÍCIO 5
Implemente um procedimento recursivo para imprimir todos os números naturais de 0 até N (lido) em ordem "decrescente".

Crie uma função recursiva que retorne a soma dos elementos de um vetor de inteiros.

EXERCÍCIO 6
Implemente uma função recursiva que calcule a soma dos primeiros n cubos: $S = 1^3 + 2^3 + ... + n^3$

EXERCÍCIO 7
Escreva uma função recursiva que receba um valor inteiro “x” e o retorne o valor invertido.
Exemplo: se x = 123, a função deve retornar 321.

OBS: Não imprimir dentro da função; Não use String.

EXERCÍCIO 8
Escreva uma função recursiva que retorne o menor elemento em um vetor.

EXERCÍCIO 9
Faça uma função recursiva que inverta os elementos de um vetor.

EXERCÍCIO 10
P*alíndromo* é uma palavra que pode ser lida, indiferentemente, da esquerda para a direita ou da direita para a esquerda. Exemplo: SERER, ARARA, OVO, ABBA. Escreva uma função recursiva que determine se uma palavra é ou não um palíndromo.

EXERCÍCIO 11
Implemente uma função recursiva que receba um valor inteiro em base decimal e o imprima em base binária.

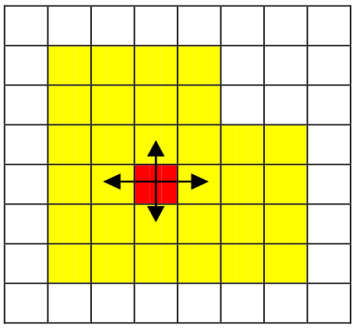
EXERCÍCIO 12
Implemente uma função recursiva que calcule a soma dos dígitos de um número inteiro. Por exemplo, se a entrada for 123, a saída deverá ser 1+2+3 = 6.

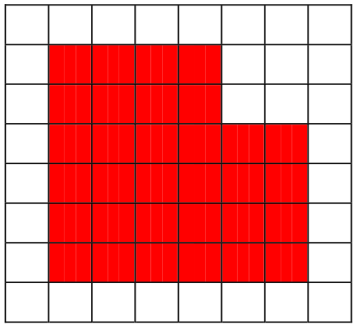
EXERCÍCIO 13
Pode-se calcular o quociente da divisão, DIV , de x por y , dois números inteiros,
usando-se a seguinte definição:

$$\mathrm{DIV}(x,y)=\begin{cases}1+\mathrm{DIV}(|x|-|y|,|y|), & se\ |x|>|y| \\ 0, & se\ |x|<|y| \\ 1, & se\ |x|=|y|\end{cases}$$

Então, pede-se que seja criada uma função recursiva para descrever tal definição. A função deve retornar -1 caso não seja possível realizar o cálculo. Além disso, crie um algoritmo que leia os dois valores inteiros e utilize a função criada para calcular o quociente de x por y , e imprima o valor computado.

EXERCÍCIO 14
Uma imagem discreta de largura w e altura h, pode ser representada em um computador através de uma matriz I[i,j], de ordem w x h, que armazena em cada posição um número inteiro entre 0 e 255, o qual especifica uma certa cor em uma paleta de cores. Em pacotes de pintura interativos é muito comum a operação que efetua o preenchimento de certa área de uma imagem com uma cor c_{ant} com uma nova cor c. Esta operação pode ser realizada de forma simples através de um método denominado *Boundary-fill*.
O procedimento em questão recebe como entrada um ponto no interior da região especificado por índices (x,y) e a cor de preenchimento c. O algoritmo inicialmente detecta a cor c_{ant} no ponto (x,y) e começa pintando tal posição com a cor c caso $c \neq c_{ant}$. O processo é repetido recursivamente para os vizinhos abaixo I[x+1,y], acima I[x-1,y], à esquerda I[x,y-1] e à direita I[x,y+1] desde que estejam dentro da imagem e possuam cor igual a c_{ant} , isto é, igual a cor a ser substituída. Escreva procedimento que implemente tal algoritmo.





EXERCÍCIO 15
Pesquise e resolva o problema das oito rainhas de forma recursiva.

EXERCÍCIO 16
Pesquise e resolva o problema do passeio do cavaleiro em um tabuleiro de xadrez de forma recursiva.

EXERCÍCIO 17
Seja um labirinto descrito através de uma matriz de booleanos, onde cada posição com valor igual a verdadeiro corresponde a uma passagem livre e uma posição falsa representa uma obstrução. Escreva um algoritmo que encontre um caminho que leve uma posição inicial qualquer a uma saída do labirinto, caso exista. Uma saída é uma posição livre na borda da matriz que define o labirinto.

Implemente uma solução para o problema das Torres de Hanói com 4 pinos.

ANÁLISE DO COMPORTAMENTO ASSINTÓTICO DE FUNÇÕES

EXERCÍCIO 18
O custo de utilização de um algoritmo pode ser medido de várias maneiras. Descreva as principais técnicas, apontando suas eventuais vantagens e desvantagens.

EXERCÍCIO 19
O que significa dizer que uma função g(n) é O(f(n))?

EXERCÍCIO 20
Indique se as afirmativas abaixo são verdadeiras ou falsas e justifique a sua resposta utilizando a definição de dominação assintótica em cada caso.

a) $2^{2n} = O(3^n)$
b) $2^{n+1} = o(3^n)$
c) $f(n) = O(u(n))$ e $g(n) = O(v(n)) \Rightarrow f(n) + g(n) = O(u(n) + v(n))$
d) $f(n) = O(u(n))$ e $g(n) = O(v(n)) \Rightarrow f(n) - g(n) = O(u(n) - v(n))$

EXERCÍCIO 21
Prove que $f(n) = 1^2 + 2^2 + \ldots + n^2$ é igual a $n^3/3 + O(n^2)$.

EXERCÍCIO 22
Apresente um algoritmo para obter o maior e o segundo maior elemento de um conjunto. Apresente também uma análise do algoritmo. Você acha o seu algoritmo eficiente? Por quê? Procure comprovar suas respostas.

EXERCÍCIO 23
Implemente em Linguagem Java os três algoritmos apresentados nos Programas 1.3, 1.4 e 2.8 do livro texto, para obter o máximo e o mínimo de um conjunto contendo n elementos. Execute os algoritmos para valores suficientemente grandes de n, gerando casos de teste para o melhor caso, pior caso e caso esperado. Meça o tempo de execução para cada algoritmo dos três casos acima. Comente os resultados obtidos.

EXERCÍCIO 24
Avalie as somas:

a) $\sum_{i=1}^{n} a^i$

b) $\sum_{i=1}^{n} \frac{1}{i}$

c) $\sum_{i=1}^{n} \log(i)$

d) $\sum_{i=1}^{n-1} \frac{1}{i(i+1)}$

EXERCÍCIO 25
Resolva as seguintes equações de recorrência:

a) $\begin{cases} T(n) = T(n-1) + c, \quad c \, constante, n > 1 \\ T(1) = 0 \end{cases}$

b) $\begin{cases} T(n) = 2T(n/4) + n, para \, n > 1 \\ T(1) = 27 \end{cases}$

c) $\begin{cases} T(n) = xT(n/2) + yn, para \, n > 1 \\ T(1) = 1 \end{cases}$

EXERCÍCIO 26
Apresente a complexidade de tempo para as funções abaixo:

```
void funcao(int a, int n) {
    for (int i=1; i < 3; i++)
        for (int j=i; j < n+1; j++)
            for (int k=i; k < j+1; k++)
                if (a % 2 == 0)
                    a = a + i + j + k;
}
```

```
int funcao(int n) {
    if (n == 0)
        return(1);
    else
        return(funcao(n-1)+funcao(n-1));
}
```

```
void pesquisa(int n) {
    if (n > 1) {
        // Inspecione n*n*n elementos;
        pesquisa(2n/3);

}
```

```
void funcao(int n) {
    if (n > 1) {
        // Divide n em 3 partes iguais
        // n1, n2, n3 com custo de n*n
        // passos;
        funcao(n1);
        funcao(n2);
        funcao(n3);
    }
}
```

BOM ESTUDO!