



Universidade Federal Fluminense
Instituto de Computação
Departamento de Ciência da Computação
Programação de Computadores II
Professor: Leandro Augusto Frata Fernandes

5ª Lista de Exercícios

Recursão

1. Usando recursividade, calcule a soma de todos os valores de um array de reais.
2. Usando recursividade encontre o maior elemento de um array de inteiros.
3. Faça uma função recursiva que inverta os elementos de um vetor.
4. Implemente uma busca ternária recursiva.
5. Implemente uma solução para o problema das Torres de Hanói com 4 pinos.
6. Implemente na forma de procedimentos os seguintes algoritmos de forma recursiva. Faça um programa que teste os procedimentos implementados.

a) Bubble sort

b) Seleção Direta

7. Um método bastante utilizado para avaliar polinômios da forma $p_n(x) = a_0x^n + a_1x^{n-1} + a_2x^{n-2} + \dots + a_{n-1}x + a_n$ é o que se baseia na *Regra de Horner*. Um método iterativo baseado nesta regra segue o seguinte esquema:

$$b_0 = a_0$$

$$b_{i+1} = x \cdot b_i + a_{i+1}; i = 0, 1, \dots, n-1$$

através do qual pode-se obter $b_n = p_n(x)$.

Uma alternativa consiste em escrever $p_n(x) = x \cdot p_{n-1}(x) + a_n$, onde $p_{n-1}(x) = a_0x^{n-1} + a_1x^{n-2} + \dots + a_{n-2}x + a_{n-1}$, que é uma formulação recursiva para o esquema. Escreva uma versão iterativa e uma versão recursiva de um algoritmo para avaliação de polinômios.

8. Um palíndromo é uma palavra que é igual a si mesma lida de trás para frente (exemplos: rotor, arara, etc.). Escreva um algoritmo recursivo que determine se uma palavra é um palíndromo.

9. Escreva um algoritmo recursivo capaz de gerar todos os elementos do conjunto potência dado conjunto formado por letras.

Exemplo: $X = \{a, b, c\}$ então $2^X = \{\emptyset, \{a\}, \{b\}, \{c\}, \{a, b\}, \{a, c\}, \{b, c\}, \{a, b, c\}\}$

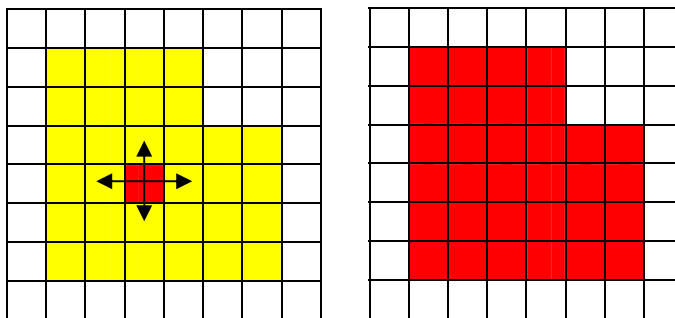
10. Escreva um algoritmo recursivo capaz de gerar todas as permutações de uma string formada por letras que não se repetem.

11. Uma imagem discreta de largura w e altura h , pode ser representada em um computador através de uma matriz $I[i, j]$, de ordem $w \times h$, 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 acima $I[x+1,y]$, abaixo $I[x-1,y]$, 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.



12. Resolva o problema das oito rainhas de forma recursiva.
13. Resolva o problema do passeio do cavaleiro em um tabuleiro de xadrez de forma recursiva.
14. 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.