

UFMS - DEPARTAMENTO DE COMPUTAÇÃO E ESTATÍSTICA
Algoritmos e Estruturas de Dados II
Prof. Marco Aurélio

LISTA 3 – ENTREGA ATÉ 02/06 ÀS 19H NA SECRETARIA DO DCT

Observação: você deve apresentar seus algoritmos utilizando o pseudocódigo visto em sala como linguagem.

1. Considere uma lista linear encadeada. Escreva um procedimento para executar cada uma das seguintes operações:
 - (a) Reverta a lista, isto é, o último elemento torna-se o primeiro, e assim por diante.
 - (b) Combine duas listas ordenadas em uma única lista ordenada.
 - (c) Construa uma lista contendo a união dos elementos das duas listas.
2. Considere uma lista duplamente encadeada. Escreva um procedimento para executar cada uma das seguintes operações:
 - (a) Concatene duas listas.
 - (b) Construa uma lista contendo a interseção dos elementos das duas listas.
 - (c) Coloque os elementos da lista em ordem crescente.
3. Sejam os seguintes algoritmos de busca em uma lista ordenada de tamanho n em alocação seqüencial:

BUSCA-ORD($\mathcal{L}, n, x$)	BUSCA-ORD'($\mathcal{L}, n, x$)
01. $\mathcal{L}[n+1].chave \leftarrow x$	01. <u>se</u> $x \leq \mathcal{L}[n].chave$ <u>então</u>
02. $i \leftarrow 1$	02. $i \leftarrow 1$
03. <u>enquanto</u> $\mathcal{L}[i].chave < x$ <u>faça</u>	03. <u>enquanto</u> $\mathcal{L}[i].chave < x$ <u>faça</u>
04. $i \leftarrow i + 1$	04. $i \leftarrow i + 1$
05. <u>se</u> $i = n + 1$ <u>ou</u> $\mathcal{L}[i].chave \neq x$ <u>então</u>	05. <u>se</u> $\mathcal{L}[i].chave \neq x$ <u>então</u>
06. <u>retorne</u> -1	06. <u>retorne</u> -1
07. <u>senão</u>	07. <u>senão</u>
08. <u>retorne</u> i	08. <u>retorne</u> i
	09. <u>senão</u>
	10. <u>retorne</u> -1

- (a) Compare os algoritmos acima em termos do número total de operações realizadas e de suas complexidades de tempo.
 - (b) Em que situação o desempenho dos dois é equivalente?
 - (c) Qual é a restrição que o BUSCA-ORD' apresenta em relação ao BUSCA-ORD?
4. A memória principal pode ser vista como um grande vetor $M[n]$. Um Sistema de Alocação de Memória pode ser mantido por meio de uma lista duplamente ligada indicando os blocos alocados e os blocos livres e seus respectivos tamanhos. Quando um bloco de memória é liberado com a função $libera(pos, tam)$, onde pos é o índice do primeiro byte do bloco e tam é o tamanho do bloco, este tem dois vizinhos (exceto nas extremidades) que podem ser blocos livres ou ocupados. Se pelo menos um for livre os blocos devem ser fundidos em um só. O algoritmo $aloca(pos, tam)$ busca na lista o primeiro bloco B livre maior ou igual a tam e divide o bloco em dois: um ocupado de tamanho tam e outro livre com tamanho $|B| - tam$. Escreva os algoritmos $libera(pos, tam)$ e $aloca(pos, tam)$ de forma a manter um Sistema de Alocação Memória.

5. Uma palavra é um *palíndromo* se a seqüência de letras que a forma é a mesma, quer seja lida da esquerda para a direita ou da direita para a esquerda (ex. raíar). Escreva um algoritmo eficiente para reconhecer se uma dada palavra é um palíndromo. Escolher a estrutura de dados conveniente para representar a palavra.
6. Seja um polinômio da forma $p(x) = a_0x^n + a_1x^{n-1} + \dots + a_nx^0$. Represente $p(x)$ através de uma lista encadeada conveniente e escreva algoritmos eficientes para efetuar as seguintes operações, onde $q(x)$ é um outro polinômio.
- (a) Calcule $p(x_0)$, onde x_0 é um dado valor para x .
 - (b) Calcule $p(q(x_0))$, onde x_0 é um dado valor para x .
 - (c) $p(x) + q(x)$.
 - (d) $p(x) \cdot q(x)$.