

Estruturas de Dados II (DEIN0083) 2016.2
Curso de Ciência da Computação
1ª avaliação

Prof. João Dallyson Sousa de Almeida

Data: 07/11/2016

Aluno: _____ Matrícula:

--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--	--

Regras durante a prova:

- É vetada: a consulta a material de apoio, conversa com colega e a utilização de dispositivos eletrônicos. A não observância de algum dos itens acima acarretará a anulação da prova.

- I. (1.0pt) Considere um algoritmo que utiliza uma matriz de $n \times m$ de inteiros para armazenar elementos. Em determinado trecho é necessário buscar o maior elemento da matriz. Considere os seguintes algoritmos, para realizar tal tarefa, baseados em comparações para encontrar o maior elemento:
- (a) Busca-se o máximo de cada linha, armazenando-se os resultados em um vetor, e depois acha-se o máximo do vetor.
- (b) Busca-se o máximo de cada coluna, armazenando-se os resultados em um vetor, e depois acha-se o máximo do vetor.
- supondo-se que $m \ll n$, qual deles realiza menos comparações? Justifique

- II. (2.0pt) Sobre análise assintótica indique se cada afirmativa é verdadeira ou falsa e justifique sua resposta:

- a) $5n^2 + 3n + 2 = \Theta(n)$ b) $5n^2 + 3n + 2 = \Omega(n)$
c) $3n + 100 \log n = O(n^2)$ d) $3n + 100 \log n = O(\log n)$

- III. (1.0pt) Utilize o teorema Mestre para analisar assintoticamente as recorrências a seguir:

Teorema 4.1 (Teorema mestre)

Sejam $a \geq 1$ e $b > 1$ constantes, seja $f(n)$ uma função e seja $T(n)$ definida sobre os inteiros não negativos pela recorrência

$$T(n) = aT(n/b) + f(n),$$

onde interpretamos n/b com o significado de $\lfloor n/b \rfloor$ ou $\lceil n/b \rceil$. Então, $T(n)$ pode ser limitado assintoticamente como a seguir.

1. Se $f(n) = O(n^{\log_b a - \epsilon})$ para alguma constante $\epsilon > 0$, então $T(n) = \Theta(n^{\log_b a})$.
2. Se $f(n) = \Theta(n^{\log_b a})$, então $T(n) = \Theta(n^{\log_b a} \lg n)$.
3. Se $f(n) = \Omega(n^{\log_b a + \epsilon})$ para alguma constante $\epsilon > 0$, e se $af(n/b) \leq cf(n)$ para alguma constante $c < 1$ e para todo n suficientemente grande, então $T(n) = \Theta(f(n))$. ■

a) $T(n) = 2 * T(n/2) + n^3$	b) $T(n) = 4 * T(n/2) + n^2$
------------------------------	------------------------------

- IV. (2.0pt) Escreva uma versão estável do algoritmo de ordenação por seleção (SelectionSort).

- V. (2.0pt) Utilize o algoritmo de ordenação HeapSort para ordenar o vetor [11, 3, 2, 25, 6, 8, 1]. Apresente, passo a passo (árvore intermediária) a estrutura da Heap (MinHeap) após a construção. Apresente a solução da ordenação decrescente mostrando passo a passo (ilustrando a árvore e o vetor em cada iteração).

- VI. (2.0pt) Ordene as oito letras iniciais do seu nome, apresentando o conteúdo do vetor a cada passo intermediário utilizando os seguintes algoritmos de ordenação. Ex: JOAO DALLYSON S. ALMEIDA, ordenaria [J, O, A, O, D, A, L, L]
- a) Inseção: liste o vetor para cada elemento incluído na ordenação parcial até o momento.
 - b) Shellsort. Use 1,3,5,7 como a sequência de valores para h . Liste o vetor para cada novo valor de h , enquanto $h > 1$. Quando $h=1$, liste o vetor para cada elemento inserido na ordem parcial.
 - c) QuickSort, usando o elemento da direita como pivô. Liste o vetor para cada nova partição completada com dois ou mais elementos.
 - d) CountSort: liste o vetor para cada elemento incluído na ordenação parcial até o momento.