Faculdade Paraíso do Ceará
SI014 – Algoritmos e Estruturas de Dados II
Prof. Vicente H. F. Batista

Lista de Exercícios 1A

entregar até 28 de setembro de 2011

Questão 1. Escreva as funções a seguir em notação O.

a) $n^3 - 1$
b) $n^2 + 2\lg n$
c) $an^n + b2^n$
d) $(n-1)^n + n^{n-1}$
e) 10^9

Questão 2. Para cada par de expressões (A, B) na tabela abaixo, indique se A é O, o, Ω, ω ou Θ de B. Assinale V para *verdadeiro* ou F para *falso*.

A	B	O	o	Ω	ω	Θ
n	$1024\lg n$					
$\lg n$	$\ln n$					
$\sqrt{n}$	$n^{\operatorname{sen} n}$					
2^n	$2^{n/2}$					
$2^{\lg 8}$	$8^{\lg n}$					
$\lg n!$	$\lg n^n$					

Questão 3. Classifique as sentenças abaixo em V (*verdadeira*) ou F (*falsa*), justificando cada uma de suas respostas.

a) Se $f_1 = \Omega(g_1)$ e $f_2 = \Omega(g_2)$, então $f_1 + f_2 = \Omega(g_1 + g_2)$.
b) Se f e g são duas funções tais que $f = O(g)$ e $g = \Omega(f)$, então $f = \Theta(g)$.
c) Se a complexidade de melhor caso de um algoritmo for f, então o número de passos que este efetua, qualquer que seja a entrada, é $\Omega(f)$.
d) Se a complexidade de pior caso de um algoritmo for f, então o número de passos que o algoritmo efetua, qualquer que seja a entrada, é $\Theta(f)$.
e) A complexidade de melhor caso de um algoritmo para um certo problema é necessariamente maior do que qualquer limite inferior para o problema.

Questão 4. Uma pessoa sobe uma escada composta de n degraus com passos que podem alcançar entre 1 e 2 degraus. Forneça a equação de recorrência que representa o número de modos distintos de a pessoa subir a escada.

Questão 5. Utilize o método de substituição para demonstrar que a solução da recorrência $T(n) = T(n/2) + 1$ é $O(\lg n)$.

Questão 6. Será que é possível aplicar o método mestre à recorrência $T(n) = 4T(n/2) + n^2 \lg n$? Justifique sua resposta e forneça um limite superior assintótico para $T(n)$.

Questão 7. Considere o problema de adicionar dois números inteiros binários de n *bits*, armazenados em dois arranjos A e B com n elementos cada. A soma dos dois números deverá ser armazenada na forma binária em um arranjo C de tamanho igual a $(n+1)$. Enuncie formalmente o problema e forneça um algoritmo para somar dois inteiros. Demonstre que seu algoritmo é correto utilizando um invariante de laço e determine as complexidades de tempo e espaço.

Questão 8. Elabore um algoritmo que determine o maior e o segundo maior elemento de um arranjo A com n números inteiros. Demonstre a correção de seu algoritmo e forneça sua complexidade de tempo.

Questão 9. Seja A uma sequência de n números distintos tal que existe um inteiro $k \in \{2, 3, \ldots, n-1\}$ satisfazendo o seguinte:

$$\begin{cases} A[i-1] < A[i], & \text{para } 2 \leq i \leq k \\ A[i] > A[i+1], & \text{para } k \leq i \leq n-1. \end{cases}$$

Elabore um algoritmo para determinar $A[k]$ em tempo $O(\lg n)$.

Parte Prática

Questão 10. Implemente em C os algoritmos Ordene-por-Inserção e Ordene-por-Entrelaçamento. Compare seus tempos de execução para vetores de diferentes tamanhos. Determine, experimentalmente, o valor máximo k de n para o qual o algoritmo Ordene-por-Inserção é mais rápido. Utilize este resultado para modificar o algoritmo Ordene-por-Entrelaçamento de modo que a recursão seja interrompida quando k for atingido, momento no qual o algoritmo Ordene-por-Inserção é executado.