Algoritmos e Estruturas de Dados II - Período - 2023.1
Professor: Carlos Vinícius G. C. Lima

Primeira Lista de Exercícios

1. Escreva as seguintes funções em notação O:
   $n^2 + n\log n$; $2n^3 + 2$; $n\log n - n$; $8\log n + \sqrt{n}$; $n! + n^n$; $7n^n + 9^n$; $(n-1)^n + n^{n-1}$; 17.

2. Para cada item abaixo, responda "certo" ou "errado", justificando:

   (a) Se a complexidade de melhor caso de um algoritmo for f, então o número de passos que o algoritmo efetua, qualquer que seja a entrada, é $\Theta(f)$.

   (b) 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 $O(f)$.

   (c) A complexidade de melhor caso de um algoritmo para um certo problema P é necessariamente maior que o limite inferior de P.

3. Escreva um algoritmo que execute a seguinte tarefa: Dado um vetor não ordenado com n elementos ($n \geq 1$), encontre **o maior e o segundo maior elementos** deste vetor. Seu algoritmo deverá percorrer o vetor uma única vez.

4. Sejam $f_1, f_2, \ldots, f_n$ uma sequência de elementos definida da seguinte forma: $f_1 = 0, f_2 = 1, f_3 = 1$ e $f_j = f_{j-1} - f_{j-2} + f_{j-3}$, para $j > 3$. Elabore um algoritmo não recursivo que determine o valor de f_n. Calcule sua complexidade em função de n.

5. Determinar a expressão da complexidade de caso médio de uma busca sequencial não ordenada de n chaves em que a probabilidade de busca da chave i é o dobro da probabilidade de busca da chave $i-1$, para $i = 2, \ldots, n$. Supor também que a probabilidade de a chave procurada se encontrar na lista é igual a q.

6. Seja V um vetor binário de tamanho n. Considere um algoritmo que modifica V em exatamente uma posição por vez, ou seja, a cada passo o algoritmo troca um dos bits do vetor anterior. Como exemplo, segue uma sequência de dois possíveis passos do algoritmo em um vetor de tamanho 5:

   $$01100 \quad \rightarrow \quad 00100 \quad \rightarrow \quad 00110$$

   Seja $T(n)$ a função que expressa o número máximo de passos que este algoritmo pode executar sem que haja repetição de qualquer vetor. Podemos expressar $T(n)$ corretamente por uma relação de recorrência indicado na letra:

   (a) $T(n) = T(n-1) + n^2$.

(b) $T(n) = 2T(n-1) + T(n-2)$.

(c) $T(n) = T(n-1) + 1$.

(d) $T(n) = T(n-1) + 2T(n-2)$.

(e) $T(n) = 2T(n-1) + 1$.

Justifique.

7. Dado um vetor V com n elementos, considere o seguinte algoritmo de ordenação:

   - Seja $m(V)$ o valor do maior elemento de V.
   - Crie um novo vetor auxiliar A de tamanho $m(V)$, com todas as posições inicialmente iguais a zero.
   - Para cada ocorrência de um elemento de valor x em V, acrescente em uma unidade o valor da posição x em A.
   - Ao final do passo anterior, para cada posição i de A, faça $A[i]$ cópias do valor i em V, caso $A[i]$ seja positivo, seguindo da posição inicial à final.

   Qual a complexidade de pior caso desse algoritmo?