Análise de Algoritmos e Estruturas de Dados
2º semestre de 2024
Lista de exercícios 1

1. Usando as definições das notações assintótica Θ, O e Ω, mostre que:

   a) $1000n^2 = O(n^2)$
   b) $5n^3 + 1000n^2 = O(n^4)$
   c) $\frac{n^2}{1000} = \Omega(n^2)$
   d) $n^4 - 25n^2 = \Omega(n^3)$
   e) $2^{16}n^2 \neq \Omega(n^3)$
   f) $n^3 - 10n^2 \neq O(n^2)$
   g) $4n^3 - 300n^2 + 7000n = \Theta(n^3)$
   h) $n^2 + nlgn = \Theta(n^2)$

2. Dados dois algoritmos **A** e **B** que solucionam um mesmo problema, e suas respectivas funções $T_{\mathbf{A}}(n)$ e $T_{\mathbf{B}}(n)$ que descrevem o tempo que algoritmos levam para executar para uma entrada de tamanho n, indique e justifique qual seria o algoritmo de sua escolha no cenário em que:

   a) $T_{\mathbf{A}}(n) = \Theta(n^2)$ e $T_{\mathbf{B}}(n) = \Theta(nlgn)$.
   b) $T_{\mathbf{A}}(n) = O(n^4)$ e $T_{\mathbf{B}}(n) = \Theta(n^3)$.
   c) $T_{\mathbf{A}}(n) = O(n^5)$ e $T_{\mathbf{B}}(n) = \Omega(n^2)$.
   d) $T_{\mathbf{A}}(n) = O(n^3)$ e $T_{\mathbf{B}}(n) = \Theta(n^3)$.
   e) $T_{\mathbf{A}}(n) = \Theta(n^2lgn)$ e $T_{\mathbf{B}}(n) = \Omega(n^2lgn)$.
   f) $T_{\mathbf{A}}(n) = \Theta(n^3\sqrt{n})$ e $T_{\mathbf{B}}(n) = O(n^3lgn)$.

3. Seja $h(n) = f(n) + g(n)$. Qual a afirmação mais precisa que se pode fazer sobre $h(n)$ quando:

   a) $f(n) = \Theta(n^3lgn)$ e $g(n) = \Theta(64^{log_4 n})$.
   b) $f(n) = \Omega(n^2\sqrt{n})$ e $g(n) = O(64^{log_2 n})$.
   c) $f(n) = \Omega(n^3)$ e $g(n) = \Omega(n^5)$.
   d) $f(n) = \Omega(nlgn)$ e $g(n) = \Theta(n^2lgn)$.
   e) $f(n) = O(n^5)$ e $g(n) = O(2^{\sqrt{n}})$.

4. Escreva uma função iterativa que recebe um vetor **a** de valores inteiros e um valor **x**, e determina todos os pares (i, j) de índices tais que $a[i] + a[j] = x$.

5. Escreva uma função iterativa que recebe duas matrizes $\mathbf{A}$ (de dimensão $n \times m$) e $\mathbf{B}$ (de dimensão $m \times p$) de valores inteiros, e devolve o produto de $\mathbf{A}$ por $\mathbf{B}$.

6. Escreva uma função recursiva que recebe um valor inteiro n e devolve seu fatorial.

7. Escreva uma função recursiva que calcula o termo a_i de uma progressão aritmética de termo inicial a_0 e razão r (obs: a função a ser implementada não será a forma mais eficiente de determinar o valor do termo a_i, mas a ideia aqui é exercitar o "pensamento recursivo").

8. Escreva uma função recursiva que recebe dois valores inteiros c e n, e devolve o valor de c^n (sem recorrer ao uso de funções prontas que implementam a exponenciação).

9. Escreva uma versão iterativa e pelo menos duas recursivas (variando a forma de dividir o problema original em subproblemas) de funções para, dado um vetor $\mathbf{a}$ de valores inteiros, resolver os seguintes problemas:

   a) determinar a soma dos valores contidos em $\mathbf{a}$.
   b) determinar se um valor x está presente em $\mathbf{a}$.
   c) determinar o número de ocorrências de valores menores ou iguais a x em $\mathbf{a}$.
   d) determinar se todos os elementos de $\mathbf{a}$ são iguais.
   e) imprimir os elementos de $\mathbf{a}$ na ordem $a[0], a[1], ..., a[n-1]$.
   f) imprimir os elementos de $\mathbf{a}$ na ordem $a[n-1], a[n-2], ..., a[0]$.

10. Para cada função recursiva implementada nos exercícios 6, 7, 8 e 9, determine a profundidade máxima de recursão que cada uma atinge, e também a quantidade total de chamadas da função que são feitas. O que estas duas medidas nos dizem a respeito do tempo de execução e consumo de memória das funções?

11. Para cada um dos métodos implementados nos exercícios 4, 5, 6, 7, 8 e 9 identifique qual o parâmetro (ou conjunto de parâmetros) que determina o "tamanho da entrada" (ou seja, está associado ao volume de trabalho que o algoritmo deve executar). Em seguida faça a análise de cada método, determinando a complexidade assintótica em relação ao tempo de execução.

12. Resolva as seguintes recorrências (assuma, para o item (d), que n é uma potência de 4, e para os itens (e) e (f), que n é uma potência de 2):

   a)
   $$T(n) = \begin{cases} a, & \text{se } n = 1 \\ T(n-1) + b, & \text{se } n > 1 \end{cases} \tag{1}$$

   b)
   $$T(n) = \begin{cases} 0, & \text{se } n = 0 \\ k, & \text{se } n = 1 \\ T(n-2) + k, & \text{se } n > 1 \end{cases} \tag{2}$$

   c)
   $$T(n) = \begin{cases} 100, & \text{se } n = 1 \\ T(n-1) + 3n, & \text{se } n > 1 \end{cases} \tag{3}$$

d)

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 4T(\frac{n}{4}) + n, & \text{se } n > 1 \end{cases} \tag{4}$$

e)

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ 2T(\frac{n}{2}) + n^3, & \text{se } n > 1 \end{cases} \tag{5}$$

f)

$$T(n) = \begin{cases} 1, & \text{se } n = 1 \\ T(\frac{n}{2}) + n, & \text{se } n > 1 \end{cases} \tag{6}$$