Prova de Validação – Projeto e Análise de Algoritmos (INE410104) – 13mar2020

Programa de Pós-Graduação em Ciência da Computação – Universidade Federal de Santa Catarina

Estudante: ____

1. Em uma propriedade rural, deseja-se abastecer um conjunto de bebedouros $B = \{b_1, b_2, \ldots, b_n\}$ para os animais a partir de uma fonte de água F. Para os bebedouros e para a fonte de água, há uma função $p : B \cup \{F\} \rightarrow \mathbb{R}^2$ que corresponde a coordenada x, y em metros da posição de um bebedouro ou da fonte no terreno da propriedade. Imagine que o terreno seja retangular. Para resolver o problema, um algoritmo que receba o conjunto B, a fonte F e a função p deve ser produzido. O algoritmo deve responder qual a metragem mínima de tubulações para conectar todos os bebedouros e a fonte em uma rede de abastecimento. Considerando o exposto, pede-se:
   (a) *(1,0pt)* Que algoritmo guloso poderia ser utilizado para resolver o problema?
   (b) *(1,5pt)* Argumente sobre a parada e a correção do algoritmo selecionado na Questão 1(a) para o problema.

2. Considere um grafo $G = (V, E)$ não-dirigido e não-ponderado, no qual $V = \{v_1, v_2, \ldots, v_n\}$. O conjunto E é formado por arestas $\{v_i, v_{i+1}\}$ para todo $i \in \{1, 2, \ldots, n-1\}$, então $|E| = n - 1$. Cada vértice v_j possui um valor inteiro positivo chamado de w_j. Deseja-se desenvolver um algoritmo para encontrar um conjunto independente[1] S em G tal que $\sum_{v_j \in S} w_j$ seja o maior possível. Com base nessas informações, responda as seguintes questões:
   (a) *(1,0pt)* Determine a função de recorrência que corresponda a função objetivo (maximização) do problema acima.
   (b) *(1,5pt)* Escreva uma implementação[2] de um algoritmo de Programação Dinâmica em uma linguagem de programação. Suponha que você tenha acesso a bibliotecas de estruturas de dados eficientes.

3. Uma das formas de se calcular um determinante para uma matriz A de ordem $n \times n$ é

$$det(A) = \sum_{j=1}^{n} (-1)^{1+j} \cdot a_{1j} \cdot det(A_{-1,-j}),$$

   na qual

$$A_{-1,-2} = \begin{pmatrix} a_{21} & a_{23} & \ldots & a_{2n} \\ a_{31} & a_{33} & \ldots & a_{3n} \\ \ldots & \ldots & \ldots & \ldots \\ a_{n1} & a_{n3} & \ldots & a_{nn} \end{pmatrix}$$

   por exemplo.

   Pede-se:
   (a) *(2,0pt)* Elabore um algoritmo de divisão e conquista que resolva o determinante de uma matriz $n \times n$ utilizando qualquer linguagem de programação, considerando a forma acima.
   (b) *(0,5pt)* Determine a recorrência representando o consumo de tempo computacional do algoritmo desenvolvido nessa questão para o cálculo do determinante.

4. Considere o algoritmo a seguir. O algoritmo recebe uma matriz quadrada $n \times n$, composta por números inteiros e devolve (retorna) um valor. Assuma que a primeira chamada é dada passando os parâmetros $(M, 1, n, 1, n)$. Assuma também que a matriz passada por parâmetro é indexada com valores entre 1 e n (intervalo fechado). Além disso, considere que "max" é uma função que retorna o maior valor de um conjunto. Por questões de facilidade, assuma que n é uma potência de 2.

[1] Um conjunto independente de um grafo $G = (V, E)$ é um conjunto $S \subseteq V$ tal que para todo o par de vértices $u, v \in S$ não exista uma aresta $\{u, v\} \in E$.

[2] A complexidade de tempo determinístico computacional dessa implementação será utilizada como critério para pontuação máxima.

$\textsc{Me}(M, l_i, l_f, c_i, c_f)$

se $l_f - l_i + 1 = 4$ **então**
    $x \leftarrow -\infty$
    para $i \leftarrow l_i$ **até** l_f **faça**
        $x \leftarrow \max\{x, m_{ii}\}$
    devolve x
$q \leftarrow l_f - l_i + 1$
$k \leftarrow \frac{q}{2}$
$a \leftarrow \textsc{Me}(M, l_i, l_i + k - 1, c_i, c_i + k - 1)$
$b \leftarrow \textsc{Me}(M, l_i, l_i + k - 1, c_i + k, c_f)$
$c \leftarrow \textsc{Me}(M, l_i + k, l_f, c_i + k, c_f)$
$x \leftarrow -\infty$
para $i \leftarrow l_i$ **até** l_f **faça**
    $j \leftarrow l_f - i$
    $x \leftarrow \max\{x, m_{ij}\}$
devolve $\max\{a, b, c, x\}$

Determine:

(a) *(2,0pt)* Escreva a função de recorrência que corresponde ao consumo de tempo computacional para o algoritmo $\textsc{Me}$, considerando uma entrada de tamanho n. Determine a fórmula fechada dessa função.

(b) *(0,5pt)* Determine a complexidade do algoritmo.

Boa prova!

Formulário:

A. Teorema mestre (Cormen et al) $\quad T(n) = aT(\frac{n}{b}) + f(n) \quad a \geq 1 \quad b > 1$

(1) Se $f(n) \in O(n^{(\log_b a) - \epsilon})$ para $\epsilon > 0$ $\Rightarrow$ $T(n) \in \Theta(n^{\log_b a})$

(2) Se $f(n) \in \Theta(n^{\log_b a})$ $\Rightarrow$ $T(n) \in \Theta(n^{\log_b a} \log n)$

(3) Se $f(n) \in \Omega(n^{(\log_b a) + \epsilon})$ para $\epsilon > 0$
e se $af(\frac{n}{b}) \leq cf(n)$ para $c < 1$
e n suficientemente grande $\Rightarrow$ $T(n) \in \Theta(f(n))$

B. Logaritmos

(1) Produto: $\log_b(x \cdot y) = \log_b x + \log_b y$

(2) Quociente: $\log_b(\frac{x}{y}) = \log_b x - \log_b y$

(3) Potência: $\log_b a^n = n \log_b a$

(4) Mudança de base: $\log_b a = \frac{\log_c a}{\log_c b}$

(5) Mudança de expoente: $a^{\log_b n} = n^{\log_b a}$

(6) Inversa: $b^{\log_b n} = n, \quad n > 0$

(7) Notação: $\log_2 x = \lg x$

C. Radiciação

(1) $\sqrt[a]{x^b} = x^{(\frac{b}{a})}$

D. Séries

(1) Soma de PA: $\sum_{i=0}^{n-1} (a_1 + i \cdot r) = \frac{n(a_1 + [a_1 + (n-1)r])}{2}$

(2) Soma de PG: $\sum_{i=0}^{n-1} (a_1 \cdot q^i) = \frac{a_1(q^n - 1)}{q - 1}$

(3) Soma de PG infinita: $S_\infty = \frac{a_1}{1-q}$, para $|q| < 1$