COS841 Complexidade de Algoritmos
Prova P1

Professores: F. Marquezino, C. Figueiredo
Universidade Federal do Rio de Janeiro

25 de Outubro de 2022

> Enviar as soluções por email, em um arquivo PDF, para Franklin, com cópia para Celina, até o dia 31 de Outubro de 2022. As soluções devem ser digitadas ou manuscritas de modo legível.

1. [2½ pontos] Diga se cada uma das afirmações abaixo é verdadeira ou falsa. Justifique sua resposta com uma prova ou um contraexemplo. *Cada resposta correta soma* $1/2$ *ponto; cada resposta errada ou com justificativa errada desconta* $1/4$ *ponto; itens deixados em branco ou sem justificativa não somam nem descontam pontos.*

   (1.1) ____ Se $f(n) = O(g(n))$ então $g(n) = O(f(n))$

   (1.2) ____ Para toda função crescente $f(n)$, temos que $f(n) = O((f(n))^2)$

   (1.3) ____ Se $f(n) = O(g(n))$ então $g(n) = \Omega(f(n))$

   (1.4) ____ Seja $p(n) = \sum_{i=0}^{d} a_i n^i$, onde $a_d > 0$, um polinômio de grau d e n, e seja k uma constante. Então, se $k \geq d$ então $p(n) = O(n^k)$. E, além disso, se $k > d$ então $p(n) = o(n^k)$.

   (1.5) ____ Considere uma modificação do algoritmo Mergesort em que o array de entrada é dividido em três partes iguais em vez de duas. Esse algoritmo modificado possui complexidade de tempo $O(N^2 \log N)$ no pior caso, em que N é o tamanho do array. Suponha $N = 3^n$.

2. [2½ pontos] Para cada situação abaixo, diga se é possível aplicar o teorema mestre para encontrar uma expressão de crescimento assintótico da complexidade, e em caso afirmativo mostre essa expressão. Justifique suas respostas. Você pode supor que $T(n)$ seja constante para $n \leq 2$. *Cada resposta correta soma* $1/2$ *ponto; cada resposta errada ou com justificativa errada desconta* $1/4$ *ponto; itens deixados em branco ou sem justificativa não somam nem descontam pontos.*

   (2.1) ____ $T(n) = 2T(n/2) + n^3$

   (2.2) ____ $T(n) = T(9n/10) + n$

   (2.3) ____ $T(n) = nT(n/2) + \log n$

   (2.4) ____ $T(n) = T(\sqrt{N}) + 1$

   (2.5) ____ $T(n) = 2T(n/4) + \sqrt{n}$

3. [2½ pontos] O Problema da Maior Subsequência Crescente (Longest Increasing Subsequence) consiste em encontrar o comprimento da maior subsequência de uma dada sequência tal que todos os elementos da subsequência estão ordenados em ordem crescente. Por exemplo, a maior subsequência crescente para $11, 21, 9, 33, 21, 53, 41, 62, 87$ é de comprimento 6, dada por $11, 21, 33, 53, 62, 87$. Elabore um algoritmo usando programação dinâmica para resolver o Problema da Maior Subsequência Crescente. Explique seu raciocínio.

4. [2½ pontos] Seja um tabuleiro n por n em que $n = 2^k$, com $k >= 1$. Ou seja, n é potência e pelo menos igual a 2. O tabuleiro tem um quadrinho faltando. Queremos preencher o tabuleiro usando peças em formato de L, ou seja, com peças no formato de quadrados 2 por 2 com um quadrinho faltando. Elabore um algoritmo do tipo "dividir e conquistar" que resolva esse problema. Explique o seu raciocínio.