IC/UFF - Análise e Síntese de Algoritmos - Primeira Prova - 2011/2

Questão 1: Para calcular o máximo divisor comum (*mdc*) entre dois números naturais não nulos n e m ($n \geq m$), normalmente utilizamos o algoritmo de Euclides. A sequência a seguir exemplifica o uso do algoritmo para $n = 1234, m = 54$:

$$1234, 54, 46, 8, 6, 2, 0$$

Na sequência acima, cada termo a partir do terceiro é igual ao resto da divisão inteira entre os dois anteriores. Temos então que $mdc(1234, 54) = 2$, que é o penúltimo termo da sequência.

(a) (1,0) Escreva equações de recorrência para calcular $mdc(n, m)$. Use a notação $x \ mod \ y$ para designar o resto da divisão inteira de x por y.
(b) (1,5) Descreva um algoritmo recursivo para calcular $mdc(n, m)$. Determine sua complexidade.

Questão 2: Considere o seguinte algoritmo:

Entrada: vetor $L[1 \cdots n]$ contendo $n > 1$ elementos
$L[0] \leftarrow \quad -\infty$ (sentinela)
para $i = 2 \ldots n$ faça
    $j \leftarrow i$
    enquanto $L[j-1] > L[j]$ faça
        troque os elementos $L[j-1]$ e $L[j]$ e subtraia 1 de j

Considerando que a operação básica do algoritmo é a *troca de elementos*, responda:

(a) (1,0) Qual a complexidade de *pior* caso deste algoritmo (número de operações básicas)? Forneça uma entrada (para $n = 5$) que leva o algoritmo para o pior caso, mostrando as trocas de elementos.
(b) (1,0) Idem, para o *melhor* caso.

Questão 3: Resolva a recorrência abaixo:

$T(1) = 1,$
$T(n) = 3\,T(n-1) + 2,$ para todo $n > 1$.

(a) (1,5) Inicialmente, use o método da força bruta para descobrir uma fórmula fechada para $T(n)$.
(b) (1,5) A seguir, prove por indução que a fórmula está correta.

Questão 4: Considere o seguinte algoritmo de ordenação:

função ordena(L)
se $|L| \leq 1$ então retornar L
senão
  $x \leftarrow$ primeiro elemento de L
  $L_1 \leftarrow$ vetor formado pelos elementos de L menores do que x
  $L_2 \leftarrow$ vetor formado pelos elementos de L iguais a x
  $L_3 \leftarrow$ vetor formado pelos elementos de L maiores do que x
  retornar a concatenação dos vetores: ordena(L_1), L_2, ordena(L_3)

(a) (1,0) Mostre a árvore de recursão deste algoritmo para a entrada $L = [5, 2, 1, 7, 9, 3, 1, 6]$. Cada nó desta árvore é uma chamada recursiva. Ao definir os vetores L_1, L_2, L_3, respeitar a mesma ordem relativa que os elementos têm em L.
(b) (1,5) Analise a complexidade de pior caso deste algoritmo para um vetor L com n elementos. (A operação básica a ser considerada no cálculo é a comparação entre dois elementos.) Mostre uma entrada com $n = 8$ que leva o algoritmo ao pior caso.