· U C ·

UNIVERSIDADE DE COIMBRA
Faculty of Science and Technology
Department of Informatics Engineering

Laboratório de Programação Avançada

First Written Test – April 19 2017

Name: ______________________________ Student ID: _______________

4 grade points in total, 1 hour and 30 minutes, closed books.

1. Derive the computational time complexity of the following recursive algorithm with respect to the number of elements in list A and justify your answer with the Master Theorem. Assume that A is a list of $n > 0$ integers, the first index of A is 1, and each arithmetic operation takes a constant amount of time. (1 g.p.)

Function $product(A, n)$
  if $n = 1$ **then**
    return
  for $i = 1$ **to** $n/2$ **do**
    $A[i] = A[i] \times A[i + (n+1)/2]$
  $product(A, (n+1)/2)$

Master Theorem (general version):
Let $a \geq 1$, $b > 1$, $d \geq 0$.

$$T(n) = \begin{cases} aT(n/b) + n^c & \text{if } n > 1 \\ d & \text{if } n = 1 \end{cases} \Rightarrow$$

$$T(n) = \begin{cases} \Theta(n^c) & \text{if } \log_b a < c \\ \Theta(n^c \log n) & \text{if } \log_b a = c \\ \Theta(n^{\log_b a}) & \text{if } \log_b a > c \end{cases}$$

2. Consider the following recursive algorithm to compute the arithmetic mean of $n > 0$ elements in a list L. Assume that the first index of list L is 1 and its elements are nonnegative reals.

Function $mean(L, n)$
  if $n = 1$ **then**
    return $L[n]$
  else
    return $L[n]/n + mean(L, n-1) \times (n-1)/n$

Show by induction that the algorithm is correct, using the mathematical definition of arithmetic mean. Explicitly state the base case, the inductive hypothesis and the inductive step. (1 g.p.)

3. Consider the following problem: Given a sequence of $n > 0$ integers, compute a contiguous subsequence that has the largest sum. For instance, for the sequence

$$(-2, 1, -3, 4, -1, 2, 1, -5, 4)$$

a contiguous subsequence with the largest sum is $(4, -1, 2, 1)$ with a value of 6. The following dynamic programming algorithm solves the problem for a sequence A of n elements by reporting only the largest sum.

Function $msum(A)$
  $DP[1] = A[1]$
  for $i = 2$ **to** n **do**
    $DP[i] = \max(A[i], DP[i-1] + A[i])$
  return $\max(DP[1], \ldots, DP[n])$

(a) Show that the problem has optimal substructure as explored by the algorithm above. (1 g.p.)

(b) Write an algorithm to retrieve a contiguous subsequence with the largest sum from the array *DP* returned by the algorithm above. (1 g.p.)