Data Structures - Assignment no. 1, 15/5, 2008

Remarks:

- **Write both your name and your ID number very clearly on the top of the exercise. Write your exercises in pen, or in clearly visible pencil. Please write *very* clearly.**
- **Give correctness and complexity proofs for every algorithm you write.**
- **For every question where you are required to write pseudo-code, also explain your solution in words.**

1. Which of the following statements are true and which are false? Only give an answer, you do not have to explain:

   (a) $34n^{10} = o(2^n)$.
   (b) $4 \log n = O(8 \log n)$.
   (c) $n^3 + \log n = \Omega(n)$.
   (d) $5^n/50 = O(n)$.
   (e) $n + 1 = O(2^n)$.
   (f) $n + 1 = o(n)$.
   (g) $n/\log n = \Theta(\log n)$.

2. Find the order of growth for the following recursively given functions. Explain your answer. Assume that $T(n)$ is constant for $n \leq 2$.

   (a) $T(n) = 8T(n/2) + 3n^2$
   (b) $T(n) = 2T(\sqrt{n}) + 1$
   (c) $T(n) = T(9n/10) + n$
   (d) $2T(n/4) + \sqrt{n}$
   (e) $T(n) = T(n-1) + 1/n$.
   (f) $T(n) = 2T(n/2) + n \log^4 n$ (Hint: The approach to solving this is similar to the approach to solving $T(n) = 2T(n/2) + n$).

3. Compute the following sums (using Θ notation).

   (a) $\sum_{k=1}^{n} 1/k$.
   (b) $\sum_{h=1}^{\lfloor \log n \rfloor} (h/2^h)$. Hint: first prove that $h \leq (1.5)^h$.

(c) $\sum_{k=1}^{\lceil \log n \rceil} c^k$.

4. The running time of an algorithm A is described by the recurrence $T(n) = 7T(n/2) + n^2$. A competing algorithm A' has a running time of $T'(n) = \mathbf{a}T'(n/4) + n^2$. What is the largest integer value for a such that A' is asymptotically faster than A?

5. Let L be a singly-linked list that consists of n elements. Each element contains a pointer "next". Someone possibly made a mistake, and the final element of the list, instead of having a "null" pointer, might be pointing to an element inside the list. What does the following pseudo-code do? What is its worst-case time complexity?

INPUT: x, which is a pointer to the first element of the list
$y \leftarrow x.next$
IF ($y == NULL$) THEN RETURN "true"
WHILE ($y \neq x$)
    $y \leftarrow y.next$
    IF ($y == NULL$) THEN RETURN "true"
    $y \leftarrow y.next$
    IF ($y == NULL$) THEN RETURN "true"
    $x \leftarrow x.next$
END WHILE
RETURN "false"