Exam - Algorithmics

Answer the following questions the best you can. Be concise and to the point: irrelevant stuff does not count. **Please write legibly and with large characters**. *Alotted time:* **1h 45min.**

1. Dorel the Clumsy Coder wants to implement MERGESORT but doesn't quite rememember how the procedure MERGE works. He does the following:

   - He recursively sorts the two halves.
   - He allocates a new array B for the merging process and copies the first sorted half into it (in that order).
   - For every element x in the second half, he linearly searches over B, from left to right, for the position p where x can be inserted. He makes space for x by shifting all elements larger than x one cell to the right (by copying them), then inserts x into B at position p.

   What is the complexity of Dorel's MERGE procedure ? (You don't need to write pseudocode if you can explain - in words - your answer). What about the overall complexity of Dorel's MERGESORT ?

2. Given the following functions $f(n) = \log_2(n)$, $g(n) = n/\log_n(n)$, $h(n) = n$, $i(n) = 1000$, which of the following statements is true ? There may be several of them. Explain your answers.

   (a) $f = O(g)$. (b) $f = o(h)$. (c) $g = o(h)$ (d) $g = \theta(h)$. (e) $g/f = \theta(i)$.

3. A sequence of number is *doubling* if every term is at least twice as large as its predecessor. For instance, in the sequence $\mathbf{1}, \mathbf{2}, 3, \mathbf{4}, 5, 6, 7, \mathbf{9}$ the bolded sequence is a doubling subsequence of length 4. Give an algorithm that takes as input a sequence of numbers and computes the length of the longest doubling subsequence, as well as a doubling sequence of the maximal length.

4. Several courses $C_1, C_2, \ldots, C_m$ may need a lab room during a given day. A course C_i is specified by a starting time s_i and an endtime t_i, $s_i < t_i$. Two courses cannot use the same room at the same time. In addition, a 15 minute break must exist between two consecutive courses in the same room (so that the room can be cleaned/set up). Give an algorithm to compute the largest number of courses that can be scheduled in the room and **prove that your algorithm finds the optimal solution**.

5. Let $a_1, a_2, \dots, a_n$ be a list of n integers in the range 1 to 1.000.000. Write the pseudocode of an algorithm for computing the shortest interval $[a, b]$ that contains at least 50% of the numbers. You need not write the pseudocode of well-known algorithms you use as subroutines. What is the complexity of your algorithm ?

6. Give an $O(n \log n)$ algorithm that takes as input a list of n numbers and a target difference x and finds two numbers a_i, a_j $(1 \leq i, j \leq n)$ such that $a_i - a_j = x$, or reports that no such numbers exist.

7. Remember, binary search on a sorted list of numbers works by dividing the list into two halves and focusing the search on one of the halves, whichever is appropriate.

   Write the pseudocode for *ternary search*, the algorithm that divides a list into **three** sublists (equal in size, if possible), and continuing the search on one of the three sublists, whichever is appropriate. Analyze the complexity of your algorithm.