Midterm

Note

This is a closed-book exam. Each problem accounts for 10 points, unless otherwise marked.

Problems

1. Consider the geometric series: 1, 2, 4, 8, 16, ... Prove *by induction* that any positive integer can be written as a sum of distinct numbers from this series.

2. Consider a round-robin tournament among n players. In the tournament, each player plays once against all other $n-1$ players. There are no draws, i.e., for a match between A and B, the result is either A beat B or B beat A. Prove *by induction* that, after a round-robin tournament, it is always possible to arrange the n players in an order $p_1, p_2, \cdots, p_n$ such that p_1 beat p_2, p_2 beat p_3, $\cdots$, and p_{n-1} beat p_n. (Note: the "beat" relation, unlike "$\geq$", is not transitive.)

3. Consider the following variant of Euclid's algorithm for computing the greatest common divisor of two positive integers.

   Algorithm Euclid_Simplified (m, n);
   begin
       // assume that $m > 0 \wedge n > 0$
       $x := m$;
       $y := n$;
       while $x \neq 0 \wedge y \neq 0$ **do**
           if $x < y$ **then** swap(x,y);
           $x := x - y$;
       od
       ...
   end

   where swap(x,y) exchanges the values of x and y.

(a) To speak about the values of a variable at different times during an execution, let m', n', x', and y' denote respectively the new values of m, n, x, and y after the next iteration of the while loop (m, n, x, and y themselves denote the current values of these variables at the start of the next iteration). Please give a precise relation between m', n', x', and y' and m, n, x, and y. (2 points)

(b) Prove *by induction* that the following is a loop invariant of the while loop:

$$x \geq 0 \wedge y \geq 0 \wedge (x \neq 0 \vee y \neq 0) \wedge \gcd(x, y) = \gcd(m, n).$$

(8 points)

4. Consider bounding summations by integrals.

   (a) If $f(x)$ is monotonically *increasing*, then

$$\int_0^n f(x)dx \leq \sum_{i=1}^{n} f(i).$$

   Show that this is indeed the case.

   (b) If $f(x)$ is monotonically *decreasing*, then

$$\sum_{i=1}^{n} f(i) \leq f(1) + \int_1^n f(x)dx.$$

   Show that this is indeed the case.

5. Show all intermediate and the final AVL trees formed by inserting the numbers 4, 5, 6, 1, 2, and 3 (in this order) into an empty tree. Please use the following ordering convention: the key of an internal node is larger than that of its left child and smaller than that of its right child. If re-balancing operations are performed, please also show the tree before re-balancing and indicate what type of rotation is used in the re-balancing.

6. The input is a set S with n real numbers. Design an $O(n)$ time algorithm to find a number that is *not* in the set. Prove that $\Omega(n)$ is a lower bound on the number of steps required to solve this problem.

7. Let $x_1, x_2, \cdots, x_{2n-1}, x_{2n}$ be a sequence of $2n$ real numbers. Design an algorithm to partition the numbers into n pairs such that the maximum of the n sums of pair is minimized. It may be intuitively easy to get a correct solution. You must explain how the algorithm can be designed using induction.

8. Apply the Quicksort algorithm to the following array. Show the contents of the array after each partition operation. If you use a different partition algorithm (from the one discussed in class), please describe it.

1	2	3	4	5	6	7	8	9	10	11	12
10	9	4	7	12	6	8	2	1	11	5	3

9. Consider rearranging the following array into a max heap using the *bottom-up* approach.

1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
7	3	5	1	2	6	14	8	11	4	10	13	15	9	12

Please show the result (i.e., the contents of the array) after a new element is added to the current collection of heaps (at the bottom) until the entire array has become a heap.

10. Prove that the sum of the heights of all nodes in a complete binary tree with n nodes is at most $n-1$. You may assume it is known that the sum of the heights of all nodes in a *full* binary tree of height h is $2^{h+1} - h - 2$. (Note: a single-node tree has height 0.)