Problem Set 1

This problem set is due in class on Thursday, September 23rd. (Monday, September 27 for CVN students.)

Each problem should be solved on a separate sheet (or sheets) of paper. Mark the top of each sheet with your name, the problem number, and the date.

When a problem asks to give an algorithm, in your solution: (i) describe shortly and informally the main ideas in your solution; (ii) give a detailed description of the algorithm, using a style similar to (but possibly more concise than) the pseudo-code of CLR; (iii) prove the correctness of the algorithm; (iv) prove a bound on the time complexity of the algorithm. You can omit the proof of correctness if it is clear from the description of the algorithm.

Problem 1. Recurrence relations

Solve the following recurrence relations ($c$ is a constant).

(a) $T(n) = 5 \cdot T\left(\frac{n}{4}\right) + cn^2$
(b) $T(n) = 3 \cdot T\left(\frac{n}{2}\right) + cn$
(c) $T(n) = 27 \cdot T\left(\frac{n}{3}\right) + cn^3$
(d) $T(n) = 2 \cdot T\left(\frac{n}{2}\right) + \sqrt{n}$
(e) $T(n) = 3 \cdot T\left(\frac{n}{3}\right) + cn^2$

Problem 2. Divide and Conquer

(a) In an infinite array the first $n$ cells contain integers in sorted order and the rest of the cells are filled with $\infty$. Describe an algorithm that takes $x$ as an input and finds the position of $x$ in the array in $O(\log n)$. You are not given the value of $n$. (If you are disturbed by the fact that the array has infinite length, assume that the array $A$ is of length $n$, and the implementation of the array data-type in your programming language is such that when accessing $A[i]$ with $i > n$ the error message $\infty$ is returned.)
(b) Given an array $S$ with the first $n$ cells containing integers in unsorted order (you are given $n$), describe an $O(n \log n)$-time algorithm that given an integer $x$ determines whether or not there exist two elements in $S$ whose sum is exactly $x$.

Problem 3. Selection

(a) Given an array $A$ containing $n$ (where $n$ is an even integer), not necessarily distinct integers, in unsorted order. Describe a linear time algorithm that determines if there exists an element $x$ that appears in $A$ at least $n/2$ times.

(b) Let $S$ be a (not necessarily sorted) sequence of $n$ integers. An integer $k$ in $S$ is said to be an approximate median of $S$ if

i) $|\{k' \in S : k' < k\}| \geq \frac{n}{4}$

ii) $|\{k' \in S : k' > k\}| \geq \frac{n}{4}$

Devise an $O(n)$ time algorithm to find all the approximate medians of $S$. 