Collaboration is allowed, but you must write your own solutions. Not all problems need divide-and-conquer approach.

**Problem 1**

Design an $O(n)$ time algorithm that given a sequence $(a_1, a_2, \ldots, a_n)$ of $n$ distinct integers and an integer $k$, $1 \leq k \leq n$, finds the $k^{th}$ smallest integer in the sequence (i.e. $k^{th}$ element from the beginning if the $n$ integers were sorted in increasing order). Clearly state and analyze the recurrence relation that you may use.

Note: In particular when $k = \lceil \frac{n}{2} \rceil$, the algorithm finds the median.

*Hint: Use a modification of the algorithm presented in class to find the median.*

**Problem 2**

Assuming that only equality checks are allowed, design an $O(n)$ time algorithm to check if there is an element which occurs more than $\frac{n}{2}$ times in an array containing $n$ elements. Note that the elements are not necessarily integers and the only operation allowed is checking whether two elements are equal.

**Problem 3**

Suppose $a > b > 1$ and $c > 0$ are constants and $T(n)$ is a function (taking non-negative values) that satisfies:

$$T(n) \leq a \cdot T\left(\frac{n}{b}\right) + cn, \quad T(1) \leq c.$$  

Show that $T(n) = O(n \log_a c)$. *Hint: Unroll the recursion in terms of $T\left(\frac{n}{b}\right), T\left(\frac{n}{b^2}\right), T\left(\frac{n}{b^3}\right), \ldots$.***

**Problem 4**

An interval $[a, b]$ is the set of all real numbers between (and including) $a$ and $b$. Given $n$ intervals, $[a_1, b_1], [a_2, b_2], \ldots, [a_n, b_n]$, design an $O(n \log n)$ time algorithm to decide whether there exists a pair of intervals that overlap (i.e. share a point).
Problem 5
Given a $m \times n$ matrix of integers such that every row is strictly increasing (from left to right), and every column is strictly increasing (from top to bottom), design an $O(m + n)$ time algorithm to test if a given integer $b$ is contained in the matrix.

Problem 6
Given a sequence of positive integers $(a_1, a_2, \ldots, a_n)$, design an $O(n)$ time algorithm to find a shortest sub-sequence of consecutive integers $(a_i, a_{i+1}, \ldots, a_j)$ whose sum is at least a given integer $M$. In other words, you want to find indices $1 \leq i \leq j \leq n$ so as to minimize $j - i + 1$ subject to the condition that $\sum_{k=i}^{j} a_k \geq M$.

(Optional, do not submit) Problem 7
A rectangle in plane is a set of the form $[a, b] \times [c, d]$. Given $n$ rectangles, design an $O(n \log n)$ time algorithm to decide whether there exists a pair of rectangles that overlap (i.e. share a point).