Problem 1. Recall that, in merge sort, the merging step combines two sorted lists $A_1, A_2$ into one sorted list $A$. Suppose that $A_1$ and $A_2$ are $(1, 8, 17, 23, 35, 83)$ and $(3, 5, 15, 28, 56)$, respectively. Describe the content of $A$ right before the moment when $23$ enters $A$.

Problem 2. For the $k$-selection problem, suppose that the input is an array of 12 elements: $(58, 23, 98, 83, 32, 24, 18, 45, 85, 91, 2, 34)$. Recall that our randomized algorithm first selects a number $v$, and then recursively solves the problem on a smaller array. Suppose that $v = 34$ and $k = 10$. What is the smaller array that we will recurse into?

Problem 3 (Textbook Exercise 9.3-5). The median of a set $S$ of $n$ elements is the $\lfloor n/2 \rfloor$ smallest element in $S$. Suppose that you are given a deterministic algorithm for finding the median of $S$ (stored in an array) in $O(n)$ worst-case time. Using this algorithm as a black box, design another deterministic algorithm for solving the $k$-selection problem (for any $k \in [1, n]$) in $O(n)$ worst-case time.

Problem 4. Let $S$ be a set of $n$ integers, and $k_1, k_2$ arbitrary integers satisfying $1 \leq k_1 \leq k_2 \leq n$. Suppose that $S$ is given in an array. Give an $O(n)$ expected time algorithm to report all the integers whose ranks in $S$ are in the range $[k_1, k_2]$. Recall that the rank of an integer $v$ in $S$ equals the number of integers in $S$ that are at most $v$.

Problem 5. Let $S$ be a set of $n$ integers given in an array. Describe a deterministic algorithm to find the 100 largest integers in $S$ in $O(n)$ worst-case time.