Algorithms: Solutions 7

Problem 1
Write an algorithm for finding the kth smallest element of a given array.

The following algorithm uses the same PARTITION procedure as QUICK-SORT. The average-case time of this algorithm is $\Theta(n)$, but the worst case is $\Theta(n^2)$.

SELECT(A, n, k)
$p \leftarrow 1$
$r \leftarrow n$
while $p < r$
    do $q =$ PARTITION(A, p, r)
        if $k \leq q$
            then $r \leftarrow q$
            else $p \leftarrow q + 1$
return $A[k]$

You may find a more sophisticated algorithm in Section 9.3 of the textbook; the worst-case time of the textbook algorithm is $\Theta(n)$.

Problem 2
Suppose that $A[1..n]$ is an array of integer numbers, and some value k occurs at least $\lfloor n/2 \rfloor + 1$ times in this array. Write an efficient algorithm for finding this value.

The "frequent" element is the median of the array; that is, it is the $\lfloor n/2 \rfloor$-th smallest element. We can find it using the SELECT algorithm from Problem 1, with $k = \lfloor n/2 \rfloor$.