Problem 1
Write an algorithm $\text{kth}(k, A[0, ..., n - 1])$ that finds the $k$th-largest elements of $A$ in $O(n)$ time. Your algorithm should use the partition algorithm we developed for quicksort. Prove your algorithm is correct and that it runs on $O(n)$ time, assuming the partition algorithm always splits its input in half.

Problem 2
If $A[0, ..., n - 1]$ is a permutation of $0, ..., n - 1$, then we can view $A$ as a function $A : \{0, ..., n - 1\} \rightarrow \{0, ..., n - 1\}$. Write an algorithm that finds $B[0, ..., n - 1]$ such that $A \circ B$ is the identity function.

Problem 3
Suppose you have a two-dimensional matrix $A[0, ..., n - 1][0, ..., m - 1]$ such that $A[i][j] \leq A[i][j + 1]$ and $A[i][j] \leq A[i + 1][j]$ for all $i$ and $j$. Find the fastest algorithm you can to determine whether a given number $x$ occurs in $A$. 