1. Let $0 < \epsilon < 1$. Suppose that we have an array $A$ of $n$ items such that the first $n - n' \epsilon$ items are sorted. Describe an $O(n)$ time algorithm to sort $A$.

2. Give the best-case, worst-case, best-case expected and worst-case expected runtime of the following function:

   **Algorithm 1: IsSortedGuess($A, 0$)**

   **Input:** Array $A$ of $n$ nonnegative integers, integer $currmax$ which is initially set to 0
   
   **Output:** A guess on whether $A$ is sorted or not.

   1. if $n == 0$ then
      2. Return “Probably Sorted”;
   3. end
   4. $i \leftarrow random(n)$;
   5. if $A[i] \geq currmax$ then
      6. $currmax \leftarrow A[i]$;
      7. Return IsSortedGuess($A[i+1...n], currmax$);
   8. end
   9. Return “Definitely Not Sorted”;

3. We have an array $A$ of $n$ non-negative integers such that each integer is less than $k$. Give an $O(n + k)$ time preprocessing algorithm such that queries of the form “how many integers are there in $A$ that are in the range $[a, b]$?” can be answered in $O(1)$ time. Note that $a$ and $b$ are not fixed; they are parameters given to the query algorithm.