1. (GT C-4.16) Suppose we are given a sequence $S$ of $n$ elements, on which a total order relation is defined (meaning any two elements can be compared). Describe an efficient method for determining whether there are two equal elements in $S$. What is the running time of your method?

2. (GT C-4.23) Given an unordered sequence $S$ of $n$ comparable elements, describe an efficient method for finding the $\lceil \sqrt{n} \rceil$ items whose rank in an ordered version of $S$ is closest to that of the median. What is the running time of your method?

3. (GT C-4.27) Given an unsorted sequence $S$ of $n$ elements, and an integer $k$, give an $O(n \lg k)$ expected time algorithm for finding the $O(k)$ elements that have rank $\lceil n/k \rceil$, $2\lceil n/k \rceil$, $3\lceil n/k \rceil$, and so on.

4. (CLRS) Let $A$ be a list of $n$ (not necessarily distinct) integers. Describe an $O(n)$-algorithm to test whether any item occurs more than $\lceil n/2 \rceil$ times in $A$. Your algorithm should use $O(1)$ additional space.

extra credit (GT C-4.14) Suppose we are given a sequence $S$ of $n$ elements, each of which is an integer in the range $[0, n^2 - 1]$. Describe a simple method for sorting $S$ in $O(n)$ time.

*Collaboration is allowed, even encouraged, provided that the names of the collaborators are listed along with the solutions. Write up the solutions on your own.