Question 1 [20 points] Given a sorted array of distinct integers $A[1 \ldots n]$, you want to find out whether there is an index $i$ for which $A[i] = i$. Give a divide-and-conquer algorithm that runs in time $O(\log n)$.

Question 2 [20 points] You are given two sorted lists of size $m$ and $n$. Give a $O(\log m + \log n)$ time algorithm for computing the $k^{th}$ smallest element in the union of the two lists.

Question 3 [20 points] An array $A[1 \ldots n]$ is said to have a majority element if more than half of its entries are the same. Given an array, the task is to design an efficient algorithm to tell whether the array has a majority element, and, if so, to find that element. The elements of the array are not necessarily from some ordered domain like the integers, and so there can be no comparisons of the form $A[i] > A[j]$? Think of the array elements as image files, for example. However you can answer questions of the form: is $A[i] = A[j]$? in constant time. Give a linear time algorithm for this task.

Question 4 [20 points] You are given a circle of radius 1 centered at $(0, 0)$. Give a procedure to return a point $(x, y)$ uniformly at random from inside this circle. You may assume that you have access to a random number generator that returns a number in $(0, 1)$ uniformly at random with exact precision. How many calls to this random number generator do you make, in expectation, to return one such point?

Question 5 [20 points] Give a procedure to merge two binary max-heaps of depth $O(h)$ and into a single heap of depth $O(h)$ in $O(h)$ time.