In class we discussed randomized quicksort, the Chernoff-Hoeffding concentration bounds, and some occupancy problems.

1. Analyze the following algorithm, a variant of binary search, for finding a query element \( q \) in a sorted array \( A \) of size \( n \), and show that with high probability it terminates in \( O(\log n) \) steps.

Algorithm Randomized-Search: Starting with the interval \([l, h] = [1, n]\), repeatedly choose uniformly at random a pivot \( p \in [l, h] \), compare \( q \) to \( A[p] \) and update the interval to be either \([l, p - 1]\) or \([p + 1, h]\), stopping if \( A[p] = q \) or \( l > h \).

2. Let \( a_1, \ldots, a_n \) be an array of numbers in the range \([0, 1]\). Design a randomized algorithm that reads only \( O(1/\varepsilon^2) \) elements, and estimates their average within additive error \( \pm \varepsilon \). The algorithm should succeed with probability at least 90%.

Extra credit:

3. Let \( a_1, \ldots, a_n \) be again an array of numbers in the range \([0, 1]\). Now design similarly a randomized algorithm that estimates their population variance \( \frac{1}{n} \sum_i a_i^2 - (\frac{1}{n} \sum_i a_i)^2 \).

Note: population variance refers to a set of reals, while the usual word variance refers to a random variable.

Hint: Estimate each of the two terms separately using the preceding exercise.