Problem 1. Design an optimal algorithm to sort n integers in the range 0 to n^3 - 1. *Pseudo-code* is not necessary.

Problem 2. We are given n points $p_1, p_2, \ldots p_n$ in the unit circle where $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for i from 1 to n. Suppose that the points are uniformly distributed (that is, the probability of finding a point in any region of the circle is proportional to the area of that region). Design an algorithm with an optimal average-case running time to sort the n points by their distances, $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin. *Pseudo-code* won't be necessary.

Problem 3. The selection algorithm (to find the kth smallest value in a list), *Blum-Floyd-Rivest-Pratt-Trajan* algorithm described in the class (and in the book), uses columns of size 5. Assume you implement the same selection algorithm using columns of size 7, rather than 5.

(a) Exactly how far from either end of the array is the median of medians guaranteed to be. Just give the high order term. (Recall that with columns of size 5 we got $\frac{3n}{10}$.)

(b) Write down the recurrence for a Selection algorithm based on columns with 7 elements each, using (the full) bubble sort to find the median of each column. (You can ignore floors and ceilings, as we did in class.) You do not have to give the algorithm, but state where each of the terms in your recurrence comes from. (For example, you might say that the n - 1 term comes from partition.)

(c) Solve the recurrence, and give the high order term exactly.

(d) How does this value compare with what we got in class for columns of size 5?