Problem 1. In class, we learned about Quicksort algorithm with a single pivot. For this problem we will modify Quicksort to use two instead of a single pivot. We will partition the array using both pivots. The elements smaller than both are left of the smaller pivot, the elements larger than both are to the right of the larger pivot and elements in-between are in the middle.

(a) Write pseudocode for the modified partition routine.

(b) Write pseudocode for Quicksort algorithm (use the modified partition routine).

(c) Find the average number of comparisons that your partition function carries out, exactly. (Hint: You would have to find the probability of all possible pivot values and sum up the number of comparisons with the probability of it happening.)

(d) Let us assume the two pivots split the array into three equal sized partitions. Write the recurrence equation and solve it.

Problem 2. In class we used the Blum-Floyd-Rivest-Pratt-Tarjan select algorithm to find the k-th smallest value using $n/5$ blocks of size 5, each. Assume for this problem, we are using $n/3$ blocks, each of size 3.

(a) Find the number of comparisons for each step of the algorithm.

(b) Obtain the overall recurrence equation for the upper bound on the number of comparisons and solve it.

(c) What are your observations?

Problem 3. We are given n points in a unit circle, $p_i = (x_i, y_i)$, such that $0 < x_i^2 + y_i^2 \leq 1$ for $i = 1, 2, \ldots, n$. 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 average-case running time of $\theta(n)$ to sort the n points by their distances $d_i = \sqrt{x_i^2 + y_i^2}$ from the origin.