You are expected to work on the problems before coming to the lab. Discussion sessions are not meant to be a lecture. TA will guide the discussion and correct your solutions if needed. We will not release "official" solutions. If you are better prepared for discussion, you will learn more. TAs will record names of the students who actively engage in discussion and report them to the instructor. The instructor factors-in participation in the final grade as bonus points.

1. (Basic) We are given a *sorted* array $A[1 \cdots n]$. We want to find the kth smallest element of A. How much time do you need?

2. (Basic) We are given a *sorted* array $A[1 \cdots n]$ and a target number x. We want to find i such that $A[i] = x$, if such i exists. How much time do you need?

3. (Basic) Let $A[1 \cdots 9] = \langle 3, 2, 9, 0, 7, 5, 4, 8, 6 \rangle$. Illustrate the execution of Randomized-Select(A, 1, 9, 4) with Randomized-Partition replaced with Partition (see page 216). More precisely, list all calls to Randomized-Select(A, p, r, i) with p, r, i specified.

4. (Basic) We say that one function f is 'polynomially' larger than the other function g if $f = \Omega(g \cdot n^\epsilon)$ for some constant $\epsilon > 0$. Equivalently, we may say that g is polynomially smaller than f.

   (a) Is n^2 polynomially larger than n?

   (b) Is $n \log n$ polynomially larger than n?

   (c) Is $n \log^{1000} n$ polynomially larger than n?

5. (Intermediate) Show how quicksort can be made to run in $O(n \log n)$ time in the worst case. Assume that all elements are distinct.

6. (Intermediate) Suppose that you have a "black-box" worst-case linear-time median subroutine. Give a simple, linear-time algorithm that solves the selection problem for an arbitrary order statistic.

7. (Advanced) Let $X[1 \cdots n]$ and $Y[1 \cdots n]$ be two arrays, each containing n numbers already in sorted order. Give an $O(\log n)$ time algorithm to find the median of all $2n$ elements in arrays X and Y. As a warm-up, you may want to find an $O(\log^2 n)$ time algorithm.