Problem 1. Consider an array of size eight with the numbers 50, 70, 10, 20, 60, 40, 80, 30. Assume you execute quicksort using the version of partition from CLRS. Note that in this algorithm an element might exchange with itself (which counts as one exchange).

(a) Show the array after the first partition. How many comparisons and exchanges are used?
(b) Show the left side after the next partition. How many comparisons are used? How many exchanges?
(c) Show the right side after the next partition on that side. How many comparisons are used? How many exchanges?
(d) What is the total number of comparisons in the entire algorithm? What is the total number of exchanges in the entire algorithm?

Problem 2. We are going to derive the average number of moves for quicksort using a somewhat unusual partitioning algorithm. We partition on the first element. Take it out. Look for the right most element that is smaller and place it in the first position (which is the newly opened position on the left side). Look for the left most element that is larger and place it in the newly opened position on the right side. Starting from there look for the right most element that is smaller and place it in the newly opened position on the left side. Starting from there look for the left most element that is larger and place it in the newly opened position on the right side. Continue in this fashion until the pointers cross. Finally, put the partition element into the hole, which is its final position in the sorted array.

(a) Assume that the partition element ends up in position $q$.
   i. What is the probability that an element originally to the left (of position $q$) went to the right (of position $q$)?
   ii. What is the expected number of elements originally to the left that go to the right?
   iii. What is the probability that an element originally to the right went to the left?
   iv. What is the expected number of elements originally to the right that go to the left?
   v. What is the total expected number of moves (for partition)?

(b) Write a recurrence for the expected number of moves (for quicksort).
(c) Simplify the recurrence, but do not solve it.
(d) To keep the calculations simple, assume you got the following recurrence:

$$ T(n) = \frac{2}{n} \left[ \sum_{q=1}^{n-1} T(q) + q - \frac{q^2}{n} \right] $$

Use constructive induction to show that $T(n) \leq an \ln n$. Try to get a good value of $a$.

(e) Floyd’s version of heapsort makes about $n \ln g n$ moves. How does quicksort compare?