1. Recall how we identified the INSERT subtask in INSERTION SORT.
   (a) Identify further subtasks (at least two) to divide INSERT into. Also, give their input/output specifications.
   (Try developing the habit of giving specifications even when you are not specifically asked to.)
   (b) Write code for INSERTION SORT and all the subtasks. Will your subtasks be different if the underlying data structures are changed?
   (When writing code for a particular function, try to use only the subtasks identified at that level. For example, the code for INSERTION SORT should ideally use INSERT, but not the subtasks identified for INSERT.)
   (c) Give proof of correctness for each of the functions designed in part (b).
   (When writing the proof for a higher level function, try to simplify the proof by assuming that the functions implementing the subtasks are correct.)

2. Show that, for a given array, the order of comparisons performed by the following two algorithms are the same. That is, for any elements a, b, c, d in the array, if a, b are compared with each other before c, d are compared in algorithm (a), then a, b will be compared with each other before c, d get compared in algorithm (b) as well.
   (a) Iterative INSERTION SORT
   (b) Recursive INSERTION SORT (as discussed in class)

3. Rewrite the recursive INSERTION SORT function so that, on an input array of size n, it recursively sorts the last $n - 1$ elements, instead of the first $n - 1$.

4. Repeat all the subparts of Question 1 on the SELECTION SORT algorithm.