

Problem 1. Selection and insertion sort are both quadratic runtime sorting algorithms. In terms of moves and/or comparisons give at least one advantage of each one over the other.

Problem 2. One way to find the median of a list is to sort the list and then take the middle element. Assume you use Bubble sort on 7 elements. Exactly how many comparisons do you use in the worst case?

Problem 3. You can actually find the median by running a sorting algorithm and stopping early, as soon as you know the median. Assume you use Bubble sort to find the median of 7 elements, but stop as soon as you know the median. Exactly how many comparisons do you use in the worst case?

Problem 4. Selection sort finds the largest element and puts it at the end of the array. Consider a version of Selection sort that finds the two largest elements and puts both of them at the end of the array (in order).

- (a) Write the pseudo-code for this version of selection sort. Make sure that it works when the size of the array is odd.
- (b) Calculate the exact worst-case number of comparisons for the case when  $n$  is even. Show your work.