



# Programming Assignment #3

3 questions

---

1  
point

1.

## GENERAL DIRECTIONS:

Download the following text file:

QuickSort.txt

The file contains all of the integers between 1 and 10,000 (inclusive, with no repeats) in unsorted order. The integer in the  $i^{th}$  row of the file gives you the  $i^{th}$  entry of an input array.

Your task is to compute the total number of comparisons used to sort the given input file by QuickSort. As you know, the number of comparisons depends on which elements are chosen as pivots, so we'll ask you to explore three different pivoting rules.

You should not count comparisons one-by-one. Rather, when there is a recursive call on a subarray of length  $m$ , you should simply add  $m - 1$  to your running total of comparisons. (This is because the pivot element is compared to each of the other  $m - 1$  elements in the subarray in this recursive call.)

WARNING: The Partition subroutine can be implemented in several different ways, and different implementations can give you differing numbers of comparisons. For this problem, you should implement the Partition subroutine *exactly* as it is described in the video lectures (otherwise you might get the wrong answer).

## DIRECTIONS FOR THIS PROBLEM:

For the first part of the programming assignment, you should always use the first element of the array as the pivot element.

## HOW TO GIVE US YOUR ANSWER:

Type the numeric answer in the space provided.

So if your answer is 1198233847, then just type 1198233847 in the space provided without any space / commas / other punctuation marks. You have 5 attempts to get the correct answer.

(We do not require you to submit your code, so feel free to use the programming language of your choice, just type the numeric answer in the following space.)

Enter answer here

---

1  
point

2.

GENERAL DIRECTIONS AND HOW TO GIVE US YOUR ANSWER:

See the first question.

DIRECTIONS FOR THIS PROBLEM:

Compute the number of comparisons (as in Problem 1), always using the final element of the given array as the pivot element. Again, be sure to implement the Partition subroutine *exactly* as it is described in the video lectures.

Recall from the lectures that, just before the main Partition subroutine, you should exchange the pivot element (i.e., the last element) with the first element.

Enter answer here

---

1  
point