2) (10 pts) ALG (Sorting)

(a) (5 pts) Consider running an Insertion Sort on the array shown below. How many swaps will execute for the duration of the algorithm running on the array shown below? Explain how you got your answer.

35	25	15	29	39	22	19	6	21

Reasoning:

Each move is a swap, since the only moves the algorithm does are swaps.

25 moves once (since 35 is larger)
15 moves twice (since both 25, 35 are larger)
29 moves once (35 is larger)
39 doesn't move (largest)
22 moves 4 times (25, 29, 35, 39 larger)
19 moves 5 times (22, 25, 29, 35, 39 larger)
6 moves 7 times (all values larger)
21 moves 5 times (22, 25, 29, 35, 39 larger)

Total swaps = 1 + 2 + 1 + 0 + 4 + 5 + 7 + 5 = 25

Number of Swaps: **25**

Grading: 2 pts for answer, 3 pts for reasoning
        If answer is off due to arithmetic error, 4/5
        If answer is correct, but reasoning doesn't make sense 2/5
        Incorrect answer with incorrect reasoning is either 0/5 or 1/5

(b) (5 pts) List the **average case** run time of each of the following sorting algorithms, in terms of n, the number of items being sorted. (Please provide Big-Oh bounds.)

      (i) Insertion Sort $O(n^2)$

      (ii) Selection Sort $O(n^2)$

      (iii) Heap Sort $O(nlgn)$

      (iv) Merge Sort $O(nlgn)$

      (v) Quick Sort $O(nlgn)$

Grading: 1 pt each all or nothing