Jonathan Rittmayer
Mat #30001299

ADS Assignment 4
Problem #1

B) - Best case: $c_1 * (n-1) = O(n)$

- Worst case: $c_2 * \sum_{i=0}^{n-1} i = c_2 * \frac{(n-1)*n}{2} = \frac{c_2}{2} * (n^2 - n) = O(n^2)$

- Average case: $c_2 * \sum_{i=0}^{n/2} i = c_2 * \frac{(\frac{n}{2}+1)\frac{n}{2}}{2} = \frac{c_2}{2} * (\frac{n^2}{4} + \frac{n}{2}) = O(n^2)$

Due to average case counting all possible outcome. $O(n^2)$ is inclusive to both the worst & best case. As such, the average-case is also $O(n^2)$. On average this sort can even do $\frac{n}{2}$ to 0 swaps

C)

Sorting Algorithms that are stable	Unstable Sorting Algorithms
1) Insertion Sort 2) Merge Sort 3) Bubble Sort	4) Heap sort

Stable sorting simply means that the relative order of duplicate elements should not change.

1. Insertion sort: If duplicate elements are encountered, the algorithm stops sorting. swapping. Since it is the same number we can put it before the duplicate element, so it means we can keep relative order.
2. Merge Sort: if left[i] <= right[i] it is a stable sort. Because the same element in the left array should be the first to go rather than the element in the right array. While this condition holds the sorting algorithm can be stable. Thus, keeping relative order.
3. Bubble sort: While the same numbers are adjacent, they don't swap. Thus it keeps the relative order of the same numbers.
4. Heap sort: Take an array of size n and assume there are two consecutive elements in the heap. For Example:

[32, 12, 21, 11, 14, 11, 9]

The relative order of the 12 and first 11 remains, However, since 21 is greater than the second 11, the 11 gets swapped. Since this occurs, the relative order is gone.

As such, with the heap sort algorithm relative order cannot be kept.

D)

Adaptive Algorithms	Non-Adaptive Algorithms
1) Insertion Sort	3) Merge Sort
2) Bubble Sort	4) Heap Sort

1) Insertion Sort uses pre-sortedness, for example lets assume the input is already sorted. Since Insertion sort is adaptive, we compare the element chosen with the elements before it. As it is sorted, it stops the inner loop after the first comparison.

2) Bubble sort also uses pre-sortedness. Because we leave the boolean variable swapped. Due to this boolean the best case doesn't make this variable change to true it only does one inner loop and then stops.

---

3) Merge sort compares the elements whether or not the list is sorted or not. This is true even with its best-case O(nlogn).

4) Heap sort Finds a maximum element whether or not it is sorted. As such it doesn't have ability to use pre-sortedness.

Both Merge and Heap sort have adaptive versions, these will of course be considered adaptive algorithms