# Q1. [5 points] A sorting algorithm is said to be stable if numbers with the same value appear in the output array in the same order as they do in the input array. Which of the following sorting algorithms are stable: insertion sort, merge sort, heapsort, and quicksort? Give a simple scheme that makes any sorting algorithm stable. (Ex. 8.3-2)

Text, letter

Description automatically generated

# Q2. [5 points] Design an algorithm to rearrange elements of a given array of n real numbers so that all its negative elements precede all its positive elements. (Hint: You can use the partition idea like the algorithm of quicksort)

Text

Description automatically generated

# Q3. [5 points] Suppose that we were to rewrite the last for loop header in the Counting sort algorithm as for j = 1 to n Show that the algorithm still works properly. Is the modified algorithm still stable?

A page of a book with writing

Description automatically generated with medium confidence

# Q4. [5 points] Show how to sort n integers in the range 0 to n3-1 in O(n) time. (Hint. Try to put a method to use radix sort and then prove its time complexity). (Ex. 8.3-4)

A page of a book with writing

Description automatically generated with medium confidence

# Q5. [5 points] Explain why the worst-case running time for bucket sort is Ɵ(n2). What simple change to the algorithm preserves its linear average-case running time and makes its worst-case running time O(n log n)? (Ex. 8.4-2)

Text, letter

Description automatically generated