

1. Answer the following questions :

(a) What is an **in place** algorithm?

(b) Which of the following sorting algorithms are in place?

-Merge Sort, Quick Sort, Insertion Sort, Counting Sort

(c) For algorithms above that are not in place **explain the reason** why they are not in place.

(d) Explain how is it possible to have a sorting algorithm with linear running time.

2. Answer the following questions :

(a) Why counting sort **is not widely used** as merge sort even though it is a linear algorithm?

(b) Describe an algorithm that finds the number of **distinct** pairs of indices (i, j) for a given array \mathbf{A} of \mathbf{n} integers, such that :

$$\forall i, j < n, \quad j > i, \quad A[i] = A[j]$$

(Integers in the array are in a linear range in terms of n)

3. Answer the following questions :

(a) What is a **stable** sorting algorithm?

(b) Why did IBM change the regular radix sort algorithm? What was the undesired property of the regular radix sort?

(c) Explain why IBM Radix sort wouldn't work if the **intermediate** sorting algorithm is not stable.

- (d) Create a scenario where counting sort is **not linear**.
- (e) Suppose we want to sort n numbers whose range is in between $[1, n^3]$. Explain how we can use IBM Radix Sort with counting sort as an intermediate sorting algorithm to sort this array in **linear time**.