Week-4 Exam :

**Nonuniform data:**  
Write a client that generates test data that is not uniform, including the following:  
Case 1: Half the data is 0s, half 1s.  
Case 2: Half the data is 0s, half the remainder is 1s, half the remainder is 2s, and so forth.   
Case 3: Half the data is 0s, half random int values. Develop and test hypotheses about the effect of such input on the performance of the algorithms in this section.

**Approach:**

The client generates an array of size 1000 for the given three cases and tests the runtime for sorting the array.

Selection Sort:

|  |  |  |
| --- | --- | --- |
| CASES | Runtime | Runtime if size doubled |
| 1 | 0.018 | 0.029 |
| 2 | 0.016 | 0.035 |
| 3 | 0.017 | 0.039 |

Insertion sort:

|  |  |  |
| --- | --- | --- |
| CASES | Runtime | Runtime if size doubled |
| 1 | 0.031 | 0.044 |
| 2 | 0.015 | 0.031 |
| 3 | 0.014 | 0.037 |

Observation:

Insertion sort and selection sort are two sorting algorithms used to sort a collection of data. Sometimes it is necessary to arrange data in a specific order. Sorting algorithms are mechanisms to sort a set of data. The insertion sort is the sorting algorithm that sorts the array by shifting elements one by one. The selection sort is the sorting algorithm that finds the smallest element in the array and exchanges the element with the first position, then find the second smallest element and exchange it with the element in the second position and continues the process till the entire array is sorted. The key difference between the insertion sort and selection sort is that insertion sort compares two elements at a time while the selection sort selects the minimum element from the whole array and sorts it.