Compare Bubble sort and Selection sort based on the following:

1. Number of comparisons

2. Number of swaps

3. Inplace and Outplace implementations

Ans – Bubble Sort -

- Sorting algorithms can be used for collections of numbers, strings, characters, or a structure of any of these types.
- Bubble sort is based on the idea of repeatedly comparing pairs of adjacent elements and then swapping their positions if they exist in the wrong order.
- Time Complexity
- Best Case: Sorted array as input. [O(N)].
- Worst Case: [O(N2)] .
- Average Case: [O(N2)] .

  Space Complexity: A temporary variable is used in swapping [O(1)]. Hence it is an In-Place sort.

Selection Sort –

- The selection sort is based on the idea of finding the minimum or maximum element in an unsorted array and then putting it in its correct position in a sorted array.
- Selection sort selects an i-th smallest element and places at i-th position. This algorithm divides the array into two parts: sorted (left) and unsorted (right) subarray. It selects the smallest element from unsorted subarray and places in the first position of that subarray (ascending order). It repeatedly selects the next smallest element.
- Best Case: Sorted array as input. [O(N2)].
- Worst Case: [O(N2)] .
- Average Case: [O(N2)] .

Bubble Sort

- Number of Comparisons ((n)*(n-1))/ 2

Selection Sort

- Number of Comparisons ((n)*(n-1))/ 2

Sehaj

21104026

Electrical