

## Tutorial 06)-3

Q-3) explain how many sorting algorithms can be performed using an appropriate array example.

D) Bubble Sort

ex- [2, 4, 1, 5, 3, 6, 7]

[2, 1, 4, 3, 5, 6, 7]

[1, 2, 3, 4, 5, 6, 7]

PP) Selection sort

ex- [1, 2, 4, 7, 5, 3, 6]

[1, 2, 3, 7, 5, 4, 6]

[1, 2, 3, 4, 5, 6, 7]

[1, 2, 3, 4, 5, 6, 7]

1) Insertion Sort

ex: [1, 7, 4, 1, 5, 3, 6]  
[2, 4, 7, 1, 5, 3, 6] sorted as initial (order)  
[1, 2, 4, 7, 5, 3, 6] right not sorted  
[1, 2, 4, 5, 7, 3, 6] requires 6 and 7  
[1, 2, 3, 4, 5, 7, 6] has less odd  
[1, 2, 3, 4, 5, 6, 7] sorted

Q-02) Compare and contrast bubble sort and selection sort algorithms.

bubble sort	selection sort
• less efficient	more efficient
• slower	faster.
• uses item exchanging	uses item selection.
• stable algorithm	not stable algorithm.
• minimum number of swaps to sort the array	maximum number of swaps to sort the array

Q-03) What are the real-world examples of sorting.

- contact lists.
- e-commerce websites.
- music libraries.
- file explorers.
- Online forums.
- library catalogs.

Q=07) Write a function using pseudo or source codes to sort an array using bubble sort and selection sort.

bubble sort :-

```

do {
    swap = false;
    for (j = 0; j < n - 1; j++)
    {
        if (arr[j] > arr[j + 1])
        {
            temp = arr[j];
            arr[j] = arr[j + 1];
            arr[j + 1] = temp;
            swap = true;
        }
    }
} while (!swap);

```

