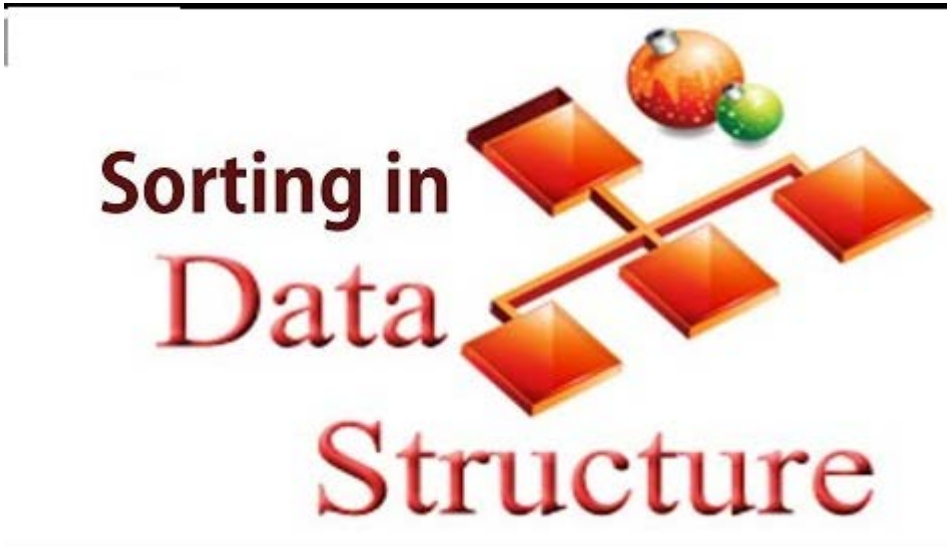

Bubble Sort

V.Savitha
AP/CSE

Sorting

- Sorting - **process of ordering** or placing a list of elements from a collection in some kind of order.
- Storage of data in **sorted** order.
- Ascending and descending order.
- Makes **searching easier**.

Sorting Techniques

- Sorting technique depends on **two parameters.**

1. Execution time of program - time taken for execution of program.
2. Space taken by the program.

Sorting techniques are differentiated by their **efficiency and space requirements.**

Sorting Techniques

Bubble Sort

Insertion Sort

Selection Sort

Quick Sort

Heap Sort

Shell Sort

Merge Sort

Bubble Sort

It compares all the elements one by one and sorts them based on their values

$O(n^2)$ time

Sort the Array using Bubble Sort

40	10	20	30	50

Starts with first two element 40 > 10, 10 is small, so swap the value

40	10	20	30	50

40 > 20, 20 is small, so swap the value

10	40	20	30	50

40 > 30, 30 is small, so swap the value

10	20	40	30	50

50 > 40, so it is already sorted

10	20	30	40	50

Sorted Array in Ascending order

10	20	30	40	50

Bubble Sort - Example

Example:

First Pass:

(**5 1** 4 2 8) –> (**1 5** 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1.

(1 **5 4** 2 8) –> (1 **4 5** 2 8), Swap since 5 > 4

(1 4 **5 2** 8) –> (1 4 **2 5** 8), Swap since 5 > 2

(1 4 2 **5 8**) –> (1 4 2 **5 8**), Now, since these elements are already in order (8 > 5), algorithm does not swap them.

Second Pass:

(**1 4** 2 5 8) –> (**1 4** 2 5 8)

(1 **4 2** 5 8) –> (1 **2 4** 5 8), Swap since 4 > 2

(1 2 **4 5** 8) –> (1 2 **4 5** 8)

(1 2 4 **5 8**) –> (1 2 4 **5 8**)

Now, the array is already sorted, but our algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Third Pass:

(**1 2** 4 5 8) –> (**1 2** 4 5 8)

(1 **2 4** 5 8) –> (1 **2 4** 5 8)

(1 2 **4 5** 8) –> (1 2 **4 5** 8)

(1 2 4 **5 8**) –> (1 2 4 **5 8**)

Bubble Sort – Algorithm

```
void BubbleSort (int a[ ], int n)
    {
            int i, temp, j;
            for (i = 1; i < = n; i ++)
            {
                    for (j = 1; j < = i; j++)
                    {
                            if (a[j] > a[j + 1])
                            {
                                    temp = a [j]
                                    a[j] = a[j + 1];
                                    a[j + 1] = temp;
                            }
                    }
            }
    }
```