BUBBLE SORT ALGORITHM

1. DEFINITION:

- Bubble Sort is an algorithm which is used to sort **N** elements that are given in a memory for eg: an Array with **N** number of elements. Bubble Sort compares all the element one by one and sort them based on their values.

- It is called Bubble sort, because with each iteration the smaller element in the list bubbles up towards the first place, just like a water bubble rises up to the water surface.

- Sorting takes place by stepping through all the data items one-by-one in pairs and comparing adjacent data items and swapping each pair that is out of order.

===

2. EXPLAINATION:

- Let us take the array of numbers "5 1 4 2 8", and sort the array from lowest number to greatest number using bubble sort. In each step, elements written in **bold** are being compared. Three passes will be required.

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 the 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**)

==

3. PSEUDOCODE IMPLEMENTATION:

```
/* Pseudocode*/
func bubbleSort
//
        for i from 0 to N
//
          swaps = 0
```

```
//
        for j from 0 to N - 2
//
              if array[j] > array[j + 1]
//
                swaps(array[j], array[j + 1])
//
                swaps = swaps + 1
//
              if swaps = 0
//
                break
end func bubbleSort
```