**MODULE - 4**

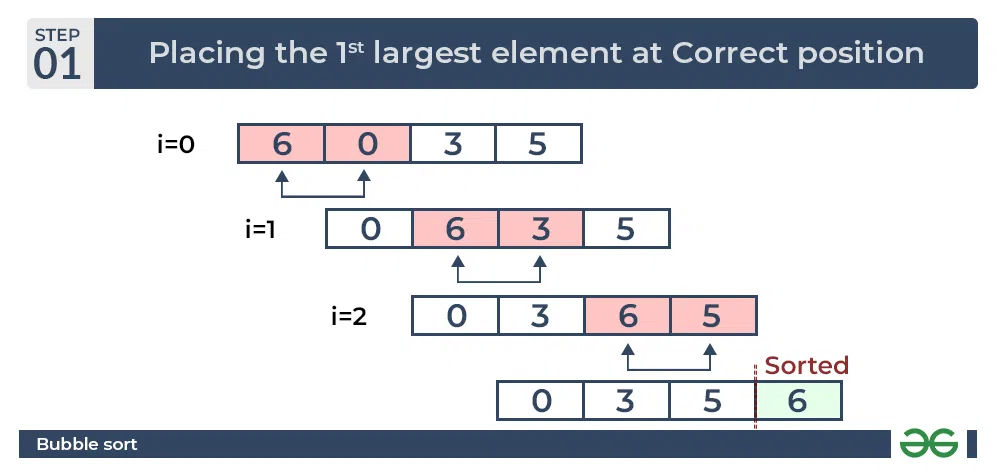
* Bubble Sort

1. Traverse from left and compare adjacent elements and the higher one is placed at right side.
2. In this way, the largest element is moved to the rightmost end at first.
3. This process is then continued to find the second largest and place it and so on until the data is sorted.

***Input:****arr[] = {6, 3, 0, 5}*

***First Pass:***

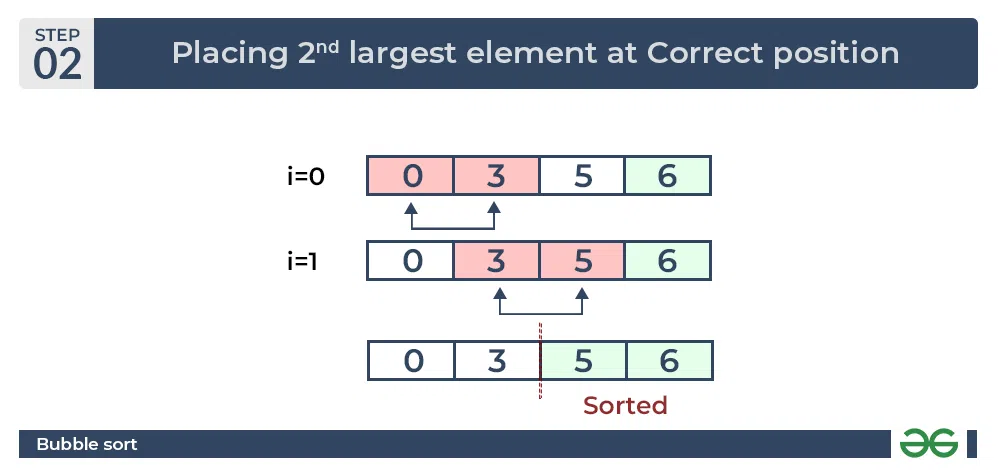
*The largest element is placed in its correct position, i.e., the end of the array.*

**

*Bubble Sort Algorithm : Placing the largest element at correct position*

***Second Pass:***

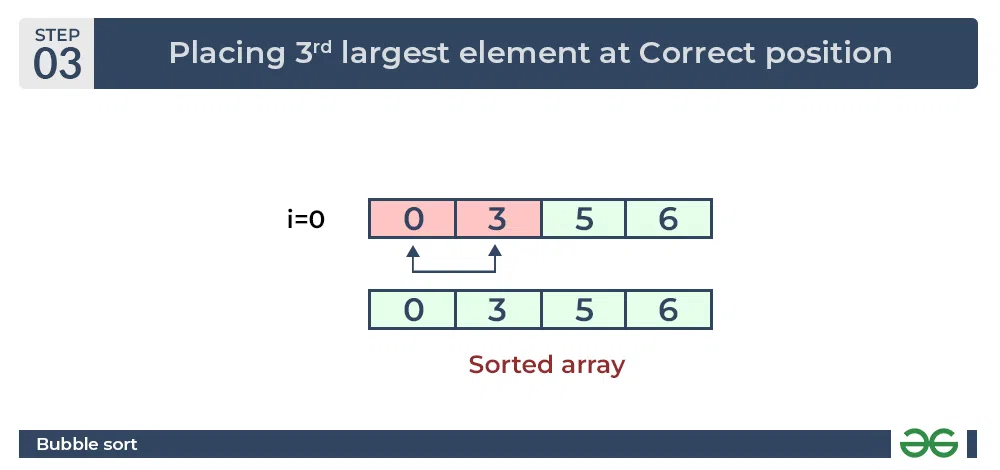
*Place the second largest element at correct position*

**

*Bubble Sort Algorithm : Placing the second largest element at correct position*

***Third Pass:***

*Place the remaining two elements at their correct positions.*

**

*Bubble Sort Algorithm : Placing the remaining elements at their correct positions*

**Total no. of passes:**n-1

**Total no. of comparisons:**n\*(n-1)/2

**Time Complexity:** O(n^2)

## ****Advantages of Bubble Sort:****

* Bubble sort is easy to understand and implement.
* It does not require any additional memory space.
* It is a stable sorting algorithm, meaning that elements with the same key value maintain their relative order in the sorted output.

## ****Disadvantages of Bubble Sort:****

* Bubble sort has a time complexity of O(N2) which makes it very slow for large data sets.
* Bubble sort is a comparison-based sorting algorithm, which means that it requires a comparison operator to determine the relative order of elements in the input data set. It can limit the efficiency of the algorithm in certain cases.

**Algorithm**

Step 1: START

Step 2: FOR all array elements

Step 3: if array[i] > array[i+1]

Step 4: SWAP(array[i], array[i+1])

Step 5: END IF

Step 6: END FOR

Step 7: END

**Code**

# include <stdio.h>

// Bubble Sort

void bubble\_sort(int arr[], int size)

{

int i,step;

for(step = 0; step < size - 1; step++)

{

for(i = 0; i < size - 1; i++)

{

if (arr[i] > arr[i+1])

{

int temp;

temp = arr[i];

arr[i] = arr[i+1];

arr[i+1] = temp;

}

}

}

}

// display array

void display(int arr[], int size)

{

int i;

for (i=0; i < size; i++)

printf("%d ", arr[i]);

}

int main()

{

int a[] = {12,544,87,13,61,564,13,19,4};

int size = sizeof(a) / sizeof(a[0]);

printf("Original Array: ");

display(a, size);

bubble\_sort(a, size);

printf("\nSorted Array: ");

// display(a, size);

for (int i = 0; i < size; i++)

{

printf("%d ", a[i]);

}

}