

Bubble sort

Bubble sort

This is a simple sorting algorithm.
The Input to this algorithm will be like this

Enter how many numbers you want to sort in
Increasing order: 6
Enter the numbers to be sorted:
1,3,5,2,4,6

The output of this algorithm will be like this:

The result after sorting your numbers in increasing order is:

1,2,3,4,5,6

The array of data to be sorted is
1,3,5,2,4,6 (in increasing order)

The process of “Bubble sorting”

$1 < 3$ no swapping



The array of data to be sorted is
1,3,5,2,4,6

The process of “Bubble sorting”

$3 < 5$ no swapping



The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

$5 > 2$ **swapping**

1 3 **2 5** 4 6



The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

5 > 4 **swapping**

1 3 2 **4 5** 6




The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

$5 < 6$ no swapping

1 3 2 4 5 6



The diagram shows the array [1, 3, 2, 4, 5, 6]. The last two elements, 5 and 6, are enclosed in a blue rectangular box. Below this box, there are two red arrows pointing upwards, one towards the number 5 and one towards the number 6, indicating the current elements being compared in the sorting process.

The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

2nd pass $1 < 3$ no swapping

1 3 2 4 5 6



The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

2nd pass $3 > 2$ swapping



The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

2nd pass $3 < 4$ no swapping

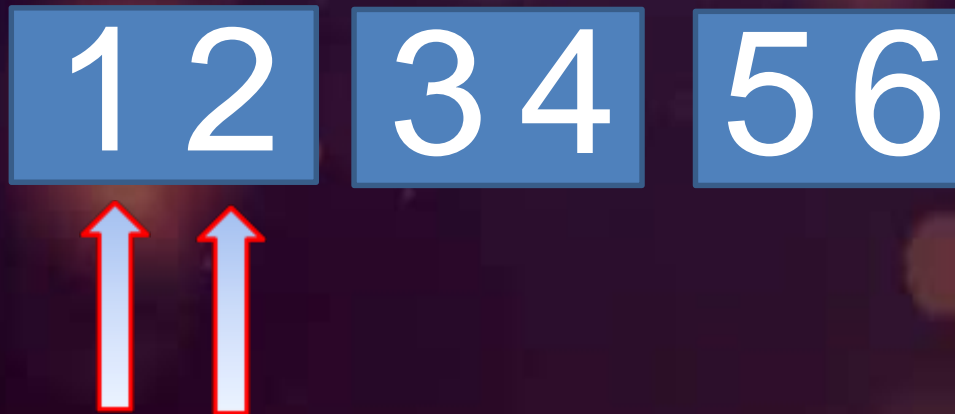


The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

3rd pass $1 < 2$ no swapping



The array of data to be sorted is

1,3,5,2,4,6

The process of “Bubble sorting”

The result after “Bubble sorting” is

1	2	3	4	5	6
---	---	---	---	---	---

ALGORITHM:

Bubble sort (int arr [], int n)

```
{  
    int i, j, temp;  
    for ( i = 0; i < n - 1; i++ ) {  
        for ( j = 0 ; j < n - 1 - i ; j++ )  
            {  
                if( arr [ j ] > arr [ j + 1 ] )  
                {  
                    temp = arr [ j ];  
                    arr [ j ] = arr [ j + 1 ];  
                    arr [ j + 1 ] = temp;  
                }  
            }  
    }  
}
```

QUESTION

Sort: 5, 4, 3, 2, 1 using bubble sort

Here: $n = 5$

Pass 1: 4, 3, 2, 1, **5**

Pass 2: 3, 2, 1, **4**, **5**

Pass 3: 2, 1, **3**, **4**, **5**

Pass 4: 1, **2**, **3**, **4**, **5**

Sort the following elements:

70, 25, 55, 65, 15, 5, 68

Pass 1 : 25, 55, 65, 15, 5, 68, 70

Pass 2 : 25, 55, 15, 5, 65, 68, 70

Pass 3 : 25, 15, 5, 55, 65, 68, 70

Pass 4 : 15, 5, 25, 55, 65, 68, 70

Pass 5 : 5, 15, 25, 55, 65, 68, 70

Pass 6 : 5, 15, 25, 55, 65, 68, 70

Important Points:

1. Number of Passes: $n - 1$, where n is the size of given array.
2. In every pass, Maximum n comparison and n swaps.
3. Total Number of Comparisons = n^2
4. Total Number of Swaps = n^2
5. Minimum number of swaps = 0 if array is already sorted
e.g. 10, 20, 30, 40, 50
6. Bubble Sort is in-place sorting because it does not require an additional array to sort given data elements.
7. Worst Case Time Complexity (O)
Best Case Time Complexity (Ω)
Average Case Time Complexity (Θ)
8. Bubble Sort is an example of **Stable Sort**.


$$2n^2 = O(n^2)$$


$$n^2$$