



# Bubble Sort

## Bubble Sort

**Bubble Sort** is a simple `sorting` algorithm that **repeatedly steps through the list, compares adjacent elements**, and swaps them if they are in the wrong order.

This process is repeated until **the array is sorted**.

After **each pass**, the largest unsorted element "bubbles up" to its correct position at the end of the array. It's called "Bubble Sort"

As smaller elements slowly "bubble" to the top of the list.

## Approach:

**Iterate** through the `array` multiple times.

In each pass, compare **adjacent elements**.

If the `current element` is **greater than the next one**, swap them.

**After each pass**, the largest unsorted element bubbles up to its correct position at the end.

Use a `boolean` flag (`isSwapped`) to **track whether any swapping happened**.

If no swaps occurred in a full pass, the `array` is already sorted → early exit (optimization).

Repeat this process for  $n - 1$  passes (where  $n$  is the array length), or until no **swaps are needed**.

## Time & Space Complexity:

**Time Complexity:  $O(n)$  (Best Case)** when array is already sorted (optimized with `isSwapped`).

**Worst Case:  $O(n^2)$**  When array is in reverse order.

**Space Complexity:  $O(1)$**  In-place sorting, no extra space used.

## Dry Run

**Input:** arr = [4, 5, 1, 3, 9]

Pass 1 (i = 0):

j = 0 → [4, 5, 1, 3, 9] → 4 < 5 → no swap

j = 1 → [4, 5, 1, 3, 9] → 5 > 1 → swap → [4, 1, 5, 3, 9]

j = 2 → [4, 1, 5, 3, 9] → 5 > 3 → swap → [4, 1, 3, 5, 9]

j = 3 → [4, 1, 3, 5, 9] → 5 < 9 → no swap

Pass 2 (i = 1):

j = 0 → [4, 1, 3, 5, 9] → 4 > 1 → swap → [1, 4, 3, 5, 9]

j = 1 → [1, 4, 3, 5, 9] → 4 > 3 → swap → [1, 3, 4, 5, 9]

j = 2 → [1, 3, 4, 5, 9] → 4 < 5 → no swap

Pass 3 (i = 2):

j = 0 → [1, 3, 4, 5, 9] → 1 < 3 → no swap

j = 1 → [1, 3, 4, 5, 9] → 3 < 4 → no swap

→ No swaps → break

**Final:** arr: [1, 3, 4, 5, 9]

JavaScript

Python

Java

C++

C

C#

```
let arr = [4,5,1,3,9]
```

```
function bubbleSort(arr){  
  let n = arr.length;  
  for(let i=0; i<n; i++) {  
    let temp=arr[i]  
    arr[i]=arr[i+1];  
    arr[i+1]=temp;  
  }  
}
```

```
        arr[j+1]=temp;
        isSwapped=true;
    }
}
if(!isSwapped)
    break;
}
return arr;
}
let result = bubbleSort(arr)
console.log("Sorted array",result)
```

**Video****Course****Discuss doubts****Contribute****Certificate**

## Bubble Sort - DSA Notes

Bubble Sort - DSA Notes