

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" because 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.

Dry Run (arr = [3, 1, 4]):

Initial State: [3, 1, 4]

Pass 1 (`i = 0`):

`j = 0`: Compare `arr[0] = 3` and `arr[1] = 1`

→ $3 > 1$ → swap

→ `arr = [1, 3, 4]`

`j = 1`: Compare `arr[1] = 3` and `arr[2] = 4`

→ $3 < 4$ → no swap

→ At least one swap was made, so continue.

Pass 2 (`i = 1`):

`j = 0`: Compare `arr[0] = 1` and `arr[1] = 3`

→ $1 < 3$ → no swap

→ No swaps made → array is sorted → exit early

Time Complexity:

Best Case: $O(n)$ → 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

```
function bubbleSort(arr) {
  let n = arr.length;

  for (let i = 0; i < n - 1; i++) {
    let swapped = false;

    for (let j = 0; j < n - 1 - i; j++) {
      if (arr[j] > arr[j + 1]) {
        // Swap
        [arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];
        swapped = true;
      }
    }

    // If no two elements were swapped in inner loop, array is sorted
    if (!swapped) break;
  }

  return arr;
}

// Example usage:
const array = [1, 2, 3, 4, 5]; // Already sorted
console.log("Sorted array:", bubbleSort(array));
```

```
function bubbleSort(arr) {
  let n = arr.length;

  for (let i = 0; i < n - 1; i++) {
    let swapped = false;

    for (let j = 0; j < n - 1 - i; j++) {
      if (arr[j] > arr[j + 1]) {
        // Swap
        [arr[j], arr[j + 1]] = [arr[j + 1], arr[j]];
        swapped = true;
      }
    }

    // If no two elements were swapped in inner loop, array is sorted
    if (!swapped) break;
  }

  return arr;
}

// Example usage:
const array = [1, 2, 3, 4, 5]; // Already sorted
console.log("Sorted array:", bubbleSort(array));
```