

Sorting Algorithms

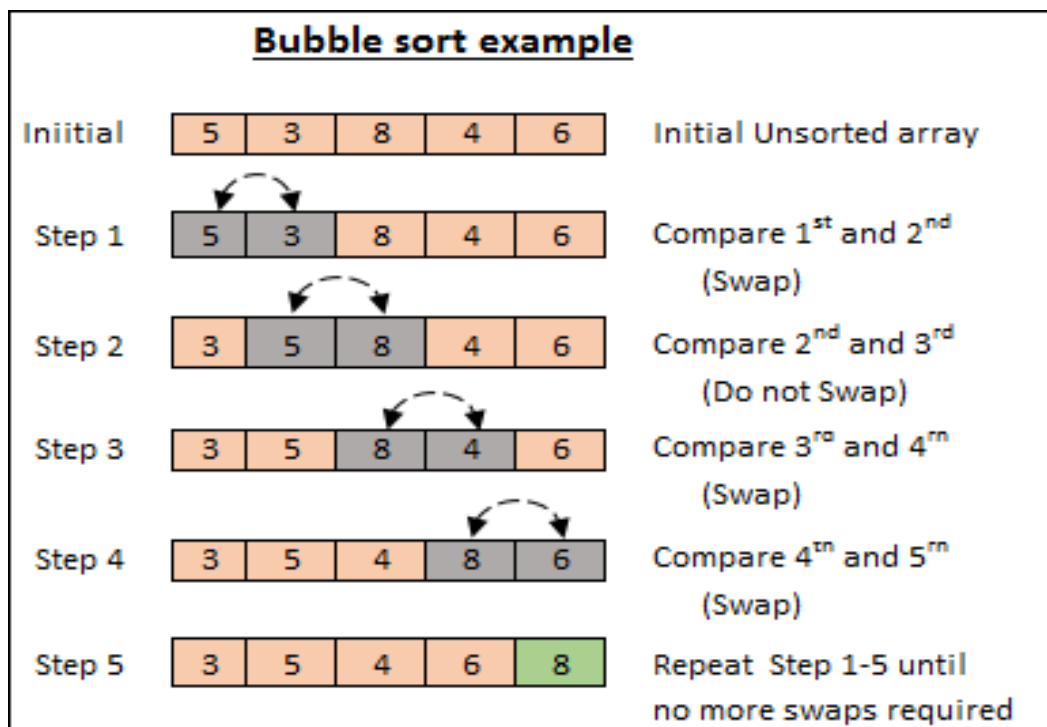
Bubble Sort

- **Description:**

- Bubble Sort is a simple sorting algorithm that repeatedly compares adjacent elements and swaps them if they are in the wrong order.
- It works by repeatedly moving the largest (or smallest) element to the end (or beginning) of the array.
- Not suitable for large data sets due to its high average and worst-case time complexity.

- **Algorithm:**

1. Traverse the array from left to right.
2. Compare adjacent elements.
3. Swap them if they are out of order.
4. Repeat until the entire array is sorted.

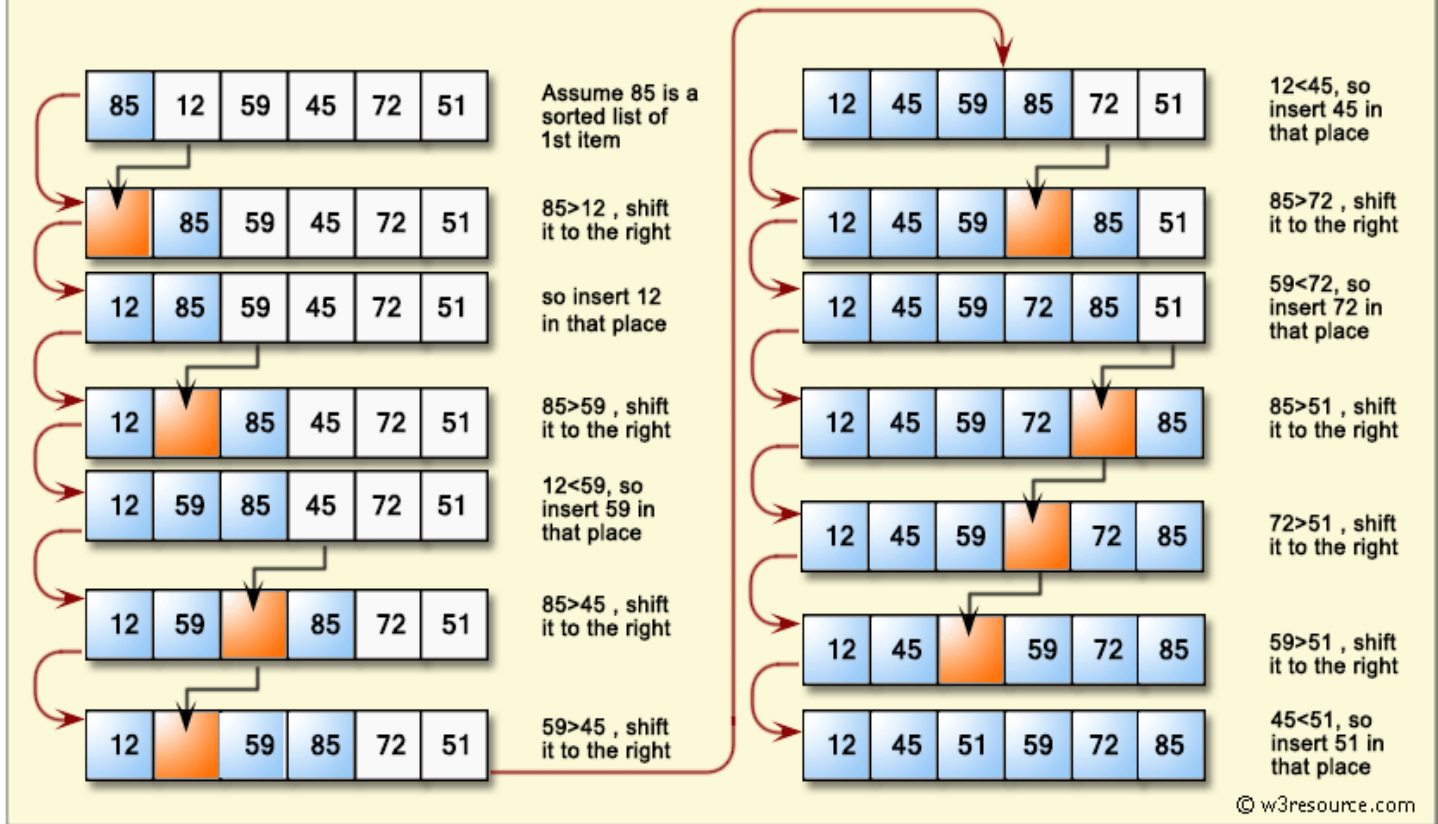


- **Complexity:**
 - Time Complexity: $O(n^2)$
 - Space Complexity: $O(1)$

Insertion Sort

- **Description:**
 - Insertion Sort builds the final sorted array one element at a time.
 - It virtually splits the array into a sorted and an unsorted part.
 - Values from the unsorted part are picked and placed in the correct position in the sorted part.
- **Algorithm:**
 1. Start with the second element (index 1) and compare it with the previous elements.
 2. If smaller, shift the previous elements to the right.
 3. Repeat until the entire array is sorted.
- **Complexity:**
 - Time Complexity: $O(n^2)$
 - Space Complexity: $O(1)$

Insertion Sort



Selection Sort

- **Description:**

- Selection Sort repeatedly selects the smallest (or largest) element from the unsorted portion of the list and moves it to the sorted portion.
- It swaps the selected element with the first element of the unsorted part.

- **Algorithm:**

1. Find the minimum (or maximum) element from the unsorted part.
2. Swap it with the first element of the unsorted part.
3. Repeat for the remaining unsorted portion.

- **Complexity:**

- Time Complexity: $O(n^2)$
- Space Complexity: $O(1)$

