

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.

The pass through the list is repeated until the list is sorted.

Here's how the Bubble Sort algorithm works:

1. Start with an unsorted list of elements.
2. Compare the first element with the second element. If the first element is greater than the second element, swap them.
3. Move to the next pair of adjacent elements and repeat the comparison and swapping process.
4. Continue this process until you reach the end of the list. At this point, the largest element will be in its correct position at the end of the list.
5. Repeat steps 2-4 for the remaining elements of the list, excluding the already sorted elements from the previous pass.
6. Repeat the above steps until the entire list is sorted.

Example to understand the Bubble Sort algorithm:

Consider the following unsorted list of integers:

```
[5, 3, 8, 2, 1].
```

Pass 1:

Compare 5 and 3. Swap them. The list becomes

[3, 5, 8, 2, 1].

Compare 5 and 8. No swap needed. The list remains

[3, 5, 8, 2, 1].

Compare 8 and 2. Swap them. The list becomes

[3, 5, 2, 8, 1].

Compare 8 and 1. Swap them. The list becomes

[3, 5, 2, 1, 8].

Pass 2:

Compare 3 and 5. No swap needed. The list remains

[3, 5, 2, 1, 8].

Compare 5 and 2. Swap them. The list becomes

```
[3, 2, 5, 1, 8].
```

Compare 5 and 1. Swap them. The list becomes

```
[3, 2, 1, 5, 8].
```

Pass 3:

Compare 3 and 2. Swap them. The list becomes

```
[2, 3, 1, 5, 8].
```

Compare 3 and 1. Swap them. The list becomes

```
[2, 1, 3, 5, 8].
```

Pass 4:

Compare 2 and 1. Swap them. The list becomes

```
[1, 2, 3, 5, 8].
```

The list is now sorted in ascending order: [1, 2, 3, 5, 8].

Bubble Sort has a time complexity of $O(n^2)$ in the average and worst cases, where 'n' is the number of elements in the list.

Bubble sort is not considered efficient for large lists but can be useful for small lists or partially sorted lists.