Bubble Sort

Bubble Sort is the simplest sorting algorithm that works by repeatedly swapping the adjacent elements if they are in the wrong order. This algorithm is not suitable for large data sets as its average and worst case time complexity is quite high.

How Bubble Sort Works?

Consider an array arr[] = {5, 1, 4, 2, 8}

First Pass:

- Bubble sort starts with very first two elements, comparing them to check which one is greater.
  - (**5 1** 4 2 8) --> (**1 5** 4 2 8), Here, algorithm compares the first two elements, and swaps since 5 > 1.
  - (1 **5 4** 2 8) --> (1 **4 5** 2 8), Swap since 5 > 4
  - (1 4 **5 2** 8) --> (1 4 **2 5** 8), Swap since 5 > 2
  - (1 4 2 **5 8**) --> (1 4 2 **5 8**), Now, since these elements are already in order (8 > 5), algorithm does not swap them.

Second Pass:

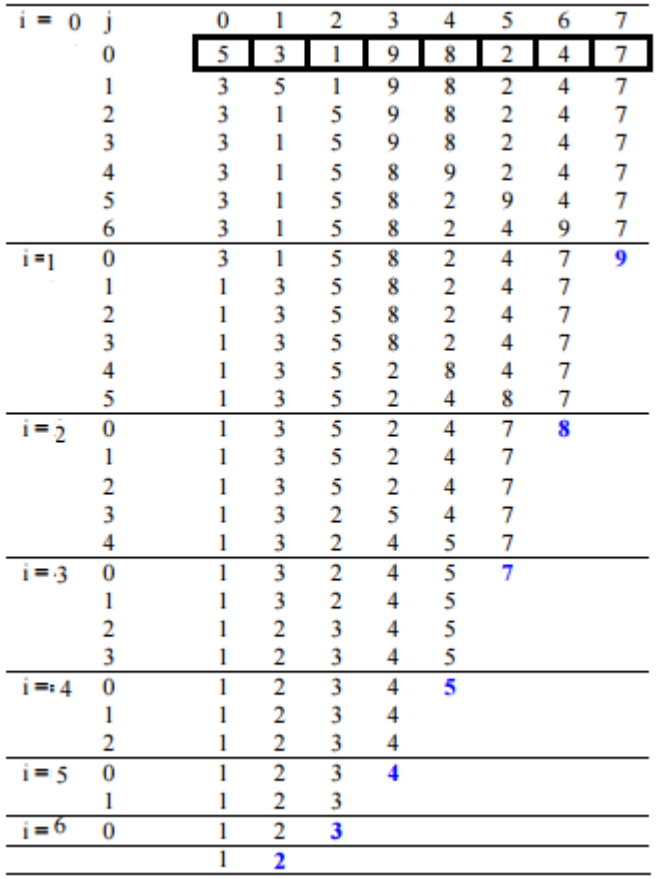
- Now, during second iteration it should look like this:

- (**1 4** 2 5 8) --> (**1 4** 2 5 8)
- (1 **4 2** 5 8) --> (1 **2 4** 5 8), Swap since 4 > 2
- (1 2 **4 5** 8) --> (1 2 **4 5** 8)
- (1 2 4 **5 8**) --> (1 2 4 **5 8**)

Third Pass:

- Now, the array is already sorted, but our algorithm does not know if it is completed.
- The algorithm needs one **whole** pass without **any** swap to know it is sorted.
  - (**1 2** 4 5 8) --> (**1 2** 4 5 8)
  - (1 **2 4** 5 8) --> (1 **2 4** 5 8)
  - (1 2 **4 5** 8) --> (1 2 **4 5** 8)
  - (1 2 4 **5 8**) --> (1 2 4 **5 8**)

Illustration:



i = 0	j	0	1	2	3	4	5	6	7
	0	5	3	1	9	8	2	4	7
	1	3	5	1	9	8	2	4	7
	2	3	1	5	9	8	2	4	7
	3	3	1	5	9	8	2	4	7
	4	3	1	5	8	9	2	4	7
	5	3	1	5	8	2	9	4	7
	6	3	1	5	8	2	4	9	7
i =1	0	3	1	5	8	2	4	7	9
	1	1	3	5	8	2	4	7	
	2	1	3	5	8	2	4	7	
	3	1	3	5	8	2	4	7	
	4	1	3	5	2	8	4	7	
	5	1	3	5	2	4	8	7	
i = 2	0	1	3	5	2	4	7	8	
	1	1	3	5	2	4	7		
	2	1	3	5	2	4	7		
	3	1	3	2	5	4	7		
	4	1	3	2	4	5	7		
i = 3	0	1	3	2	4	5	7		
	1	1	3	2	4	5			
	2	1	2	3	4	5			
	3	1	2	3	4	5			
i = 4	0	1	2	3	4	5			
	1	1	2	3	4				
	2	1	2	3	4				
i = 5	0	1	2	3	4				
	1	1	2	3					
i = 6	0	1	2	3					
		1	2						

C++ Java

```
// Java program for implementation of Bubble Sort
class BubbleSort {
    void bubbleSort(int arr[])
    {
        int n = arr.length;
        for (int i = 0; i < n - 1; i++)
            for (int j = 0; j < n - i - 1; j++)
                if (arr[j] > arr[j + 1]) {
                    // swap arr[j+1] and arr[j]
                    arr[j + 1] = temp;
                }
    }

    /* Prints the array */
    void printArray(int arr[])
    {
        int n = arr.length;
        for (int i = 0; i < n; ++i)
            System.out.print(arr[i] + " ");
        System.out.println();
    }

    // Driver method to test above
    public static void main(String args[])
    {
        BubbleSort ob = new BubbleSort();
```

```
        int arr[] = { 64, 34, 25, 12, 22, 11, 90 };
        ob.bubbleSort(arr);
        System.out.println("Sorted array");
        ob.printArray(arr);
    }
}
/* This code is contributed by Rajat Mishra */
```


Output

```
Sorted array:
1 2 4 5 8
```