Bubble Sort
Bubble Sort

• In bubble sort, the idea of the algorithm is to move higher valued elements generally towards the right and lower value elements generally towards the left.

In pseudocode:
• Set swap counter to a non-zero value
• Repeat until the swap counter is 0:
  • Reset swap counter to 0
  • Look at each adjacent pair
    • If two adjacent elements are not in order, swap them and add one to the swap counter
**Bubble Sort**

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
   Reset swap counter to 0
   Look at each adjacent pair
      If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
    Reset swap counter to 0
    Look at each adjacent pair
        If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value

Repeat until the swap counter is 0:

  Reset swap counter to 0
  Look at each adjacent pair

    If two adjacent elements are not in order, swap them and add one to the swap counter
In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
    Reset swap counter to 0
    Look at each adjacent pair
        If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
- Set swap counter to a non-zero value
- Repeat until the swap counter is 0:
  - Reset swap counter to 0
  - Look at each adjacent pair
    - If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
- Set swap counter to a non-zero value
- Repeat until the swap counter is 0:
  - Reset swap counter to 0
  - Look at each adjacent pair
    - If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
    Reset swap counter to 0
    Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
   Reset swap counter to 0
   Look at each adjacent pair
      If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
   Reset swap counter to 0
   Look at each adjacent pair
      If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
In pseudocode:

Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
   Reset swap counter to 0
   Look at each adjacent pair
      If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
In pseudocode:
Set swap counter to a non-zero value
Repeat until the swap counter is 0:
  Reset swap counter to 0
  Look at each adjacent pair
    If two adjacent elements are not in order, swap them and add one to the swap counter
Bubble Sort

• **Worst-case scenario**: The array is in reverse order; we have to “bubble” each of the $n$ elements all the way across the array, and since we can only fully bubble one element into position per pass, we must do this $n$ times.

• **Best-case scenario**: The array is already perfectly sorted, and we make no swaps on the first pass.
Bubble Sort

$O(n^2)$

$\Omega(n)$