BUBBLE SORT

ALGORITHM

- Compare the 1st and 2nd elements.
- If the 1st is larger than the 2nd, swap.
- Compare 2nd and 3rd, and swap if necessary.
- Continue comparing until the last two elements.
- The largest element is now the last element in the array.
- Repeat starting from the beginning until no swaps are performed (i.e., the array is sorted).
- Each time you go through the elements bubbling up the largest element.
- No need to try the last i elements for the ‘i’th run since the end elements are already sorted.

BUBBLE SORT

Step-by-step example

Let us take the array of numbers "5 1 4 2 8", and sort the array from lowest number to greatest number using bubble sort. In each step, the underlined are being compared. Three passes will be required.

First Pass

(**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

(**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**)
Now, the array is already sorted, but the algorithm does not know if it is completed. The algorithm needs one **whole** pass without **any** swap to know it is sorted.

Third Pass

(**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)