Bubble Sort

Mr. Poole
Java

Example - Bubble Sort

Sort the following using Bubble Sort

iteration	57	95	88	14	25	6

For each set, move the largest to the right.

Example - Bubble Sort

Sort the following using Bubble Sort

iteration	57	95	88	14	25	6

Example - Bubble Sort - Step 1

iteration	57	95	88	14	25	6
0	*57*	*88*	95	14	25	6
1	57	**88**	**95**	14	25	6
2	57	88	**14**	**95**	25	6
3	57	88	14	**25**	**95**	6
4	57	88	14	25	**6**	**95**

Like a bubble the values rise one iteration at a time.

Example - Bubble Sort - Step 2

iteration	57	88	14	25	6	95
0	*57*	*88*	14	25	6	95
1	57	**14**	**88**	25	6	95
2	57	14	**25**	**88**	6	95
3	57	14	25	**6**	**88**	95
4	57	14	25	6	*88*	*95*

Example - Bubble Sort - Step 3

iteration	57	14	25	6	88	95
0	**14**	**57**	25	6	88	95
1	14	**25**	**57**	6	88	95
2	14	25	**6**	**57**	88	95
3	14	25	6	*57*	*88*	95
4	14	25	6	57	*88*	*95*

Example - Bubble Sort - Step 4

iteration	14	25	6	57	88	95
0	**14**	**25**	6	57	88	95
1	14	**6**	**25**	57	88	95
2	14	6	*25*	*57*	88	95
3	14	6	25	*57*	*88*	95
4	14	6	25	57	*88*	*95*

Example - Bubble Sort - Step 5

iteration	14	6	25	57	88	95
0	**6**	**14**	25	57	88	95
1	6	*14*	*25*	57	88	95
2	6	14	*25*	*57*	88	95
3	6	14	25	*57*	*88*	95
4	6	14	25	57	*88*	*95*

Another Example - Bubble Sort

Example:

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 our 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)

(1 2 4 **5 8**) -> (1 2 4 **5 8**)

Code:

```
public static void bubbleSort(int [] arr){
    for(int outer = 0; outer < arr.length-1; outer++){
        for(int inner = 0; inner < arr.length-outer-1; inner++){
            if(arr[inner] > arr[inner+1]){
                int temp = arr[inner];
                arr[inner] = arr[inner+1];
                arr[inner+1] = temp;
            }
        }
    }
}
```

Swap values if left value is larger

Sorting - Bubble Sort

Runtime = O(?)

```
// A function to implement bubble sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)

    // Last i elements are already in place
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Sorting - Bubble Sort

```
// A function to implement bubble sort
void bubbleSort(int arr[], int n)
{
    int i, j;
    for (i = 0; i < n-1; i++)

    // Last i elements are already in place
    for (j = 0; j < n-i-1; j++)
        if (arr[j] > arr[j+1])
            swap(&arr[j], &arr[j+1]);
}
```

Runtime = $O(n^2)$

Because of 2 nested loops

Lab: Implement Bubble Sort

Create an array of 200 random values,
Sort them with Bubble Sort and print.