BUBBLE SORT

By Anusha Katta

What is Bubble Sort?

- In this algorithm, the elements of the list “gradually 'bubble' (or rise) to their proper location in the array.
- It sorts by comparing each pair of adjacent items and swapping them in order.
- It deals with only two elements at a time.
- The algorithm got its name from the way smaller elements “bubble” to the top of the list.

Algorithm

Step-1: Start comparing each element with its adjacent element from the starting index.

Step-2: If the current is greater than the next element, swap them.

Step-3: Repeat step 2 for all the elements of the array/list.

Step-4: Repeat steps 1, 2, and 3 (n-1) time’s for the array of length n.

Java Code

```
void bubbleSort(int arr[])
{
    int n = arr.length;
    for (int i = 0; i < n -  1; i++)
        for (int j = 0; j < n - 1; j+
        +)
            if (arr[ j] > arr[ j + 1])
                {  int temp =
                arr[ j];  arr[ j] = arr[
                j + 1];  arr[ j + 1] =
                temp;
        }
```

Bubble Sort Example

Given unsorted array = {5,1,4,2,8}

Iteration - 1:

{**5,1**,4,2,8} -> {**1,5**,4,2,8} as 5>1, 5 and 1 are swapped

{1,**5,4**,2,8} -> {1,**4,5**,2,8} as 5>4, 5 and 4 are swapped

{1,4,**5,2**,8} -> {1,4,**2,5**,8} as 5>2, 5 and 2 are swapped

{1,4,2,**5,8**} -> {1,4,2,**5,8**} as 5<8, 5 and 8 are not swapped

Bubble Sort Example continued

Iteration-2 :

{**1,4**,2,5,8} -> {**1,4**,2,5,8} as 1<4, 1 and 4 are not swapped

{1,**4,2**,5,8} -> {1,**2,4**,5,8} as 4>2, 4 and 2 are swapped

{1,2,**4,5,**8} -> {1,2,**4,5**,8} as 4<5, 4 and 5 are not swapped

{1,2,4,**5,8**} -> {1,2,4,**5,8**} as 5<8, 5 and 8 are not swapped

Bubble Sort Example continued

Iteration-3 :

{**1,2**,4,5,8} -> {**1,2,**4,5,8} as 1<2, 1 and 2 are not swapped

{1,**2,4**,5,8} -> {1,**2,4**,5,8} as 2>4, 2 and 4 are not swapped

{1,2,**4,5**,8} -> {1,2,**4,5**,8} as 4<5, 4 and 5 are not swapped

{1,2,4,**5,8**} -> {1,2,4,**5,8**} as 5<8, 5 and 8 are not swapped

Bubble Sort Example continued

Iteration-4 :

{**1,2**,4,5,8} -> {**1,2,**4,5,8} as 1<2, 1 and 2 are not swapped

{1,**2,4**,5,8} -> {1,**2,4**,5,8} as 2<4, 2 and 4 are not swapped

{1,2,**4,5**,8} -> {1,2,**4,5**,8} as 4<5, 4 and 5 are not swapped

{1,2,4,**5,8**} -> {1,2,4,**5,8**} as 5<8, 5 and 8 are not swapped

Sorted Array = {1,2,4,5,8}

Solve using Bubble Sort

Sort the below array using Bubble Sort.

Unsorted Array = {-2,45,0,11,-9}

Solve using Bubble Sort continued

Iteration - 1:

{**-2,45**,0,11,-9} -> {**-2,45**,0,11,-9} as -2<45, -2 and 45 are not swapped.

{-2,**45,0**,11,-9} -> {-2,**0,45**,11,-9} as 45>0, 45 and 0 are swapped.

{-2,0,**45,11**,-9} -> {-2,0,**11,45**,-9} as 45>11, 45 and 11 are swapped.

{-2,0,11,**45,-9**} -> {-2,0,11,**-9,45**} as 45>-9, 45 and -9 are swapped.

Solve using Bubble Sort continued

Iteration - 2:

{**-2,0**,11,-9,45} -> {**-2,0**,11,-9,45} as -2<0, -2 and 0 are not swapped.

{-2,**0,11**,-9,45} -> {-2,**0,11**,-9,45} as 0<11, 0 and 11 are not swapped.

{-2,0,**11,-9**,45} -> {-2,0,**-9,11**,45} as 11>-9, 11 and -9 are swapped.

{-2,0,-9,**11,45**} -> {-2,0,-9,**11,45**} as 11<45, 11 and 45 are not swapped.

Solve using Bubble Sort continued

Iteration - 3:

{**-2,0**,-9,11,45} -> {**-2,0**,-9,11,45} as -2<0, -2 and 0 are not swapped.

{-2,**0,-9**,11,45} -> {-2,**-9,0**,11,45} as 0>-9, 0 and -9 are swapped.

{-2,-9,**0,11**,45} -> {-2,-9,**0,11**,45} as 0<11, 0 and 11 are not swapped.

{-2,-9,0,**11,45**} -> {-2,-9,0,**11,45**} as 11<45, 11 and 45 are not swapped.

Solve using Bubble Sort continued

Iteration - 4:

{**-2,-9**,0,11,45} -> {**-9,-2**,0,11,45} as -2>-9, -2 and -9 are swapped.

{-9,**-2,0**,11,45} -> {-9,**-2,0**,11,45} as -2<0, -2 and 0 are not swapped.

{-9,-2,**0,11**,45} -> {-9,-2,**0,11**,45} as 0<11, 0 and 11 are not swapped.

{-9,-2,0,**11,45**} -> {-9,-2,0,**11,45**} as 11<45, 11 and 45 are not swapped.

Sorted Array = {-9,-2,0,11,45}

Time Complexity

- Bubble sort uses two loops- inner loop and outer loop.

Worst Case:

- It occurs when the array is in descending order.
- In worst case, both outer loop and inner loop runs O(n) times.
- Hence, the worst case time complexity of bubble sort is O(n x n) = $O(n^2)$.

Eg:- {24,17,9,5,4}

Time Complexity continued

Best Case:

- It occurs when the array is in ascending order.
- In best case, the array is already sorted but still bubble sort performs O(nxn) comparisons.
- Hence, the best case time complexity of bubble sort is $O(n^2)$.
- This could be improved by modifying the algorithm

Eg:- {2,7,9,24,58}

Average Case:

- It occurs when the array is neither ascending order nor descending order.
- In average case, bubble sort require O(n) passes for both inner and outer loop.
- Hence, the average case time complexity of bubble sort is $O(n \times n) = O(n^2)$.

Eg:- {72,45,63,1,23}

Thank You