Implement Bubble sort.

Bubble sort is a simple sorting algorithm that compares pair of adjacent terms and swaps if the order is wrong.

This sort bubbles the largest element to the end of list in first pass hence the name.

Consider input

20,19,15,25,10

19,20,15,25,10 reason of swap 20 > 19

19,15,20,25,10 reason of swap 20 > 15

19,15,20,25,10 no swaps 20 < 25

19,15,20,10,25 reason of swap 25 > 10

Now at the end of first pass we have maximum element at the end of list.

Algorithm

bubblesort(A)

Step 1 : n= length(A)

Step 2 : i=0

Step 3 : j=1

Step 4 : repeat i till n

Step 5 : repeat j till n-i

Step 6 : if A[j-1] > A[j]

Step 7 : swap(A[j-1],A[j])

Step 1: n is length of the list or array

Step 2: initialize i to 0

Step 3: initialize j to 1 as we can compare 0th and 1st element

Step 4: loop i from 0 to length of list or array

Step 5: loop j from 1 to n - i

Step 6: Check for inequality

Step 7: if inequality exists then swap the elements

swap(a,b) {

temp = a

a = b

b = temp

}

Time complexity:

Comparision can be seen in algorithm as (n-1) + (n-2) + .. + 2 + 1 =n\*(n-1)/2

So O(n^2) is worst case time complexity.

Space complexity: O(1) auxiliary space

Code for Bubble Sort:

**public** **static** **int**[] sort(**int**[] a) {

/\*\*

\* If the array is null or length is 0 then return it.

\* \*/

**if** (a == **null** || a.length == 0) {

**return** a;

}

/\*\*

\* Run outer loop for n times.

\* \*/

**for** (**int** i = 0; i < a.length; i++) {

/\*\*

\* Run inner loop for n - i times.

\* \*/

**for** (**int** j = 1; j < a.length - i; j++) {

/\*\*

\* Example: a[1] < a[0] Swap it.

\* \*/

**if** (a[j] < a[j - 1]) {

**int** temp = a[j];

a[j] = a[j - 1];

a[j - 1] = temp;

}

}

// The print statement is used to show number of passes, (i+1) is

// number of pass.

System.***out***.print("Pass " + (i + 1) + " ");

// Loop to print one pass.

**for** (**int** k = 0; k < a.length; k++) {

System.***out***.print(a[k] + " ");

}

System.***out***.println();

System.***out***.println();

}

**return** a;

}

Output:

Pass 1: 8 7 6 5 4 3 2 1 9

Pass 2: 7 6 5 4 3 2 1 8 9

Pass 3: 6 5 4 3 2 1 7 8 9

Pass 4: 5 4 3 2 1 6 7 8 9

Pass 5: 4 3 2 1 5 6 7 8 9

Pass 6: 3 2 1 4 5 6 7 8 9

Pass 7: 2 1 3 4 5 6 7 8 9

Pass 8: 1 2 3 4 5 6 7 8 9

Pass 9: 1 2 3 4 5 6 7 8 9

Also I am attaching details output i.e what happens to array after each if condition executed in inner loop.

8 9 7 6 5 4 3 2 1

8 7 9 6 5 4 3 2 1

8 7 6 9 5 4 3 2 1

8 7 6 5 9 4 3 2 1

8 7 6 5 4 9 3 2 1

8 7 6 5 4 3 9 2 1

8 7 6 5 4 3 2 9 1

8 7 6 5 4 3 2 1 9

Pass 1: 8 7 6 5 4 3 2 1 9

7 8 6 5 4 3 2 1 9

7 6 8 5 4 3 2 1 9

7 6 5 8 4 3 2 1 9

7 6 5 4 8 3 2 1 9

7 6 5 4 3 8 2 1 9

7 6 5 4 3 2 8 1 9

7 6 5 4 3 2 1 8 9

Pass 2: 7 6 5 4 3 2 1 8 9

6 7 5 4 3 2 1 8 9

6 5 7 4 3 2 1 8 9

6 5 4 7 3 2 1 8 9

6 5 4 3 7 2 1 8 9

6 5 4 3 2 7 1 8 9

6 5 4 3 2 1 7 8 9

Pass 3: 6 5 4 3 2 1 7 8 9

5 6 4 3 2 1 7 8 9

5 4 6 3 2 1 7 8 9

5 4 3 6 2 1 7 8 9

5 4 3 2 6 1 7 8 9

5 4 3 2 1 6 7 8 9

Pass 4: 5 4 3 2 1 6 7 8 9

4 5 3 2 1 6 7 8 9

4 3 5 2 1 6 7 8 9

4 3 2 5 1 6 7 8 9

4 3 2 1 5 6 7 8 9

Pass 5: 4 3 2 1 5 6 7 8 9

3 4 2 1 5 6 7 8 9

3 2 4 1 5 6 7 8 9

3 2 1 4 5 6 7 8 9

Pass 6: 3 2 1 4 5 6 7 8 9

2 3 1 4 5 6 7 8 9

2 1 3 4 5 6 7 8 9

Pass 7: 2 1 3 4 5 6 7 8 9

1 2 3 4 5 6 7 8 9

Pass 8: 1 2 3 4 5 6 7 8 9

Pass 9: 1 2 3 4 5 6 7 8 9