

Topic 3.1.1 Bubble Sort :-

Unit No. : III

1. It is exchange sort.
 2. In this we exchange elements that are out of order until the entire list is sorted.
- In bubble sort, list is divided into two sublists
Sorted & unsorted
 - Smallest element is bubbled from unsorted sublist
(largest)
and moved to sorted list.
- ie. In each pass a largest element is bubbled out to last position.

* Working :-

Topic

Unit No. :

It requires $n-1$ passes to sort data of size n .logic of operation

- $A[0]$ & $A[1]$ is compared
ie $A[0] > A[1]$ then swap the elements.
- $A[1]$ & $A[2]$ are compared
if $A[1] > A[2]$ then swap
otherwise not
- Similarly,
 $A[n-1]$ & $A[n]$ compared
& if $A[n-1] > A[n]$ then swap otherwise not
 \therefore In pass. I largest element is kept in its last position
in pass. II 2nd largest element is kept in 2nd last position

Sanjivani Rural Education Society's
College of Engineering, Kopergaon
 DEPARTMENT OF INFORMATION TECHNOLOGY

Topic

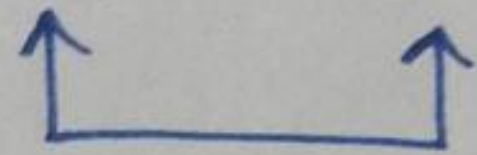
Unit No. :

Example Suppose an Array 'A' contains 8 elements.

77 33 44 11 88 22 66 55

PASS - I

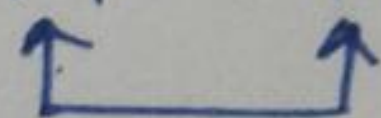
77 33 44 11 88 22 66 55



$77 > 33$

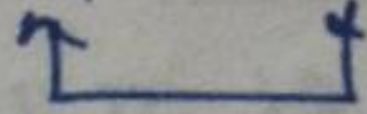
\therefore swap

33 77 44 11 88 22 66 55

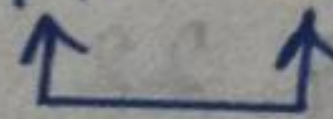


\therefore swap

33 44 77 11 88 22 66 55

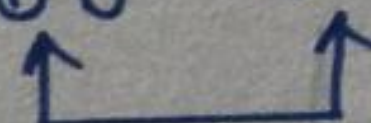


33 44 11 77 88 22 66 55



\therefore No swap

33 44 11 77 88 22 66 55



33 44 11 77 22 88 66 55



33 44 11 77 22 66 88 55



33 44 11 77 22 66 55 (88)

(88) largest element is at last position.

\therefore 33 44 11 77 22 66 55 | 88
Unsorted | sorted

Topic

Unit No. :

pass. II

33 44 11 77 22 66 55 | 88

No swap

33 44 11 77 22 66 55 | 88

Swap

33 11 44 77 22 66 55 88

No swap

swap

33 11 44 22 77 66 55 88

33 11 44 22 66 77 55 88

PASS-II 33 11 44 22 66 55 | 77 88

2nd largest
 element
 77
 is at 2nd last
 position.

PASS-III 33 11 44 22 66 55 77 88

11 33 44 22 66 55 77 88

11 33 44 22 66 55 77 88

11 33 22 44 66 55 77 88

11 33 22 44 66 55 77 88

11 33 22 44 55 | 66 77 88

Topic

Algorithm

Unit No. :

Algorithm Bubble (int a[], int n)
// sort the array by comparing Adjacent elements
4 exchanged until list is completely sorted.

1. Repeat step 2 & 3 for $k=1$ to $N-1$

2. Set $ptr=1$

3. Repeat while $ptr \leq N-k-1$

 a) If $Data[ptr] > Data[ptr+1]$ then
 Interchange $Data[ptr]$ and $Data[ptr+1]$

 b) $ptr = ptr + 1$

 end of inner loop

 end of outer loop

4. Exit

Bubble Sort (int a[], int n)

{

 int k, ptr, temp;

 for ($k=0; k \leq (N-1-1)$)

 {

 for ($j=0; j \leq N-i-1-1; j++$)

 {

 if ($a[j] > a[j+1]$)

 {

 temp = $a[j]$;

$a[j+1] = a[j]$;

$a[j] = temp$;

 }

 }

 }

}

Prepared by :