SNS COLLEGE OF ENGINEERING

Kurumbapalayam (Po), Coimbatore – 641 107

An Autonomous Institution

Accredited by NBA – AICTE and Accredited by NAAC – UGC with 'A' Grade
Approved by AICTE, New Delhi & Affiliated to Anna University, Chennai

DEPARTMENT OF COMPUTER SCIENCE AND ENGINEERING-IOT Including CS&BCT

COURSE NAME : **19CS307**- DATA STRUCTURES

II YEAR / III SEMESTER

Unit V- **SORTING AND SEARCHING**

Topic   :BUBBLE SORT

Bubble Sort

Bubble sort is a simple sorting algorithm.

This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

This algorithm is not suitable for large data sets as its average and worst case complexity are of O(n2) where n is the number of items.

Bubble Sort Working

We take an unsorted array for our example. Bubble sort takes $O(n^2)$ time so we're keeping it short and precise.

14	33	27	35	10

Bubble sort starts with very first two elements, comparing them to check which one is greater.

14	33	27	35	10

In this case, value 33 is greater than 14, so it is already in sorted locations. Next, we compare 33 with 27.

CONN….

We find that 27 is smaller than 33 and these two values must be swapped.

The   new   array   look   like   this   –

Next we compare 33 and 35. We find that both are in already sorted positions.

Then we move to the next two values, 35 and 10.

We know then that 10 is smaller 35. Hence they are not sorted.

We swap these values. We find that we have reached the end of the array. After one iteration, the array should look like this –

14	27	33	10	35

To be precise, we are now showing how an array should look like after each iteration. After the second iteration, it should look like this
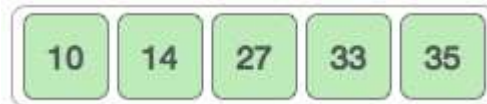
14	27	10	33	35

Notice that after each iteration, at least one value moves at the end.

14	10	27	33	35

And when there's no swap required, bubble sorts learns that an array is completely sorted.

Now we should look into some practical aspects of bubble sort.
Algorithm

We assume **list** is an array of **n** elements. We further assume that **swap** function swaps the values of the given array elements

begin BubbleSort(list)

   for all elements of list
    if list[i] > list[i+1]
      swap(list[i], list[i+1])
    end if
   end for

   return list

end BubbleSort

procedure bubbleSort(list : array of items)

   loop = list.count;

   for i = 0 to loop-1 do:
     swapped = false

     for j = 0 to loop-1 do:

       /* compare the adjacent elements */
       if list[j] > list[j+1] then
        /* swap them */
        swap(list[j], list[j+1])
        swapped = true
       end if

end for

/*if no number was swapped that means array is sorted now, break the loop.*/

if(not swapped) then
   break
end if

end for

end procedure return list

CONN….

THANK YOU