MODULE 18

SORTING OF NUMBERS USING BUBBLE SORT

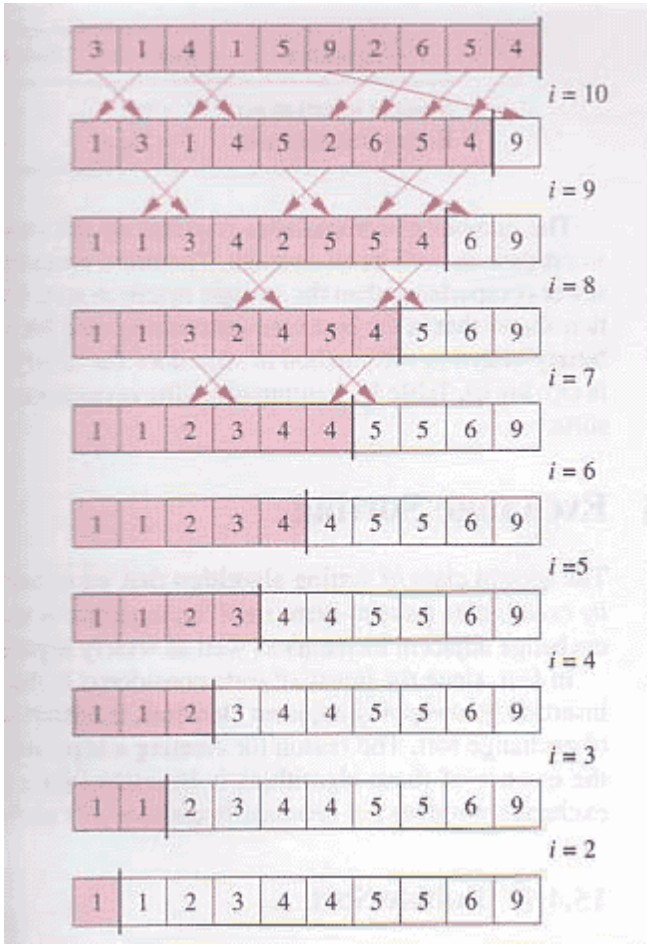
Introduction

Bubble Sort is an elementary sorting algorithm. It works by repeatedly exchanging adjacent elements, if necessary. When no exchanges are required, the file is sorted. If only one or two items in your list are out of order, bubble sort is very fast. If the items in your list are initially arranged randomly, bubble sort is extremely slow. For this reason you should be careful when you use bubble sort.

Algorithm

1. Compare each pair of adjacent elements from the beginning of an array and, if they are in reversed order, swap them.
2. If at least one swap has been done, repeat step 1.

Example. Sort {3,1,4,1,5,9,2,6,5,4} using bubble sort.



Pseudo code

Get the length of the sequence, n
Get the elements in the sequence, $a_1, a_2, \ldots, a_n$
Set the value of i to $(n - 1)$
while $i \geq 1$ **do**
  Set the value of j to 1
  while $j \leq i$ **do**
    CE(a_j, a_{j+1})
    Add 1 to the value of j
  Subtract 1 from the value of i

Hints to solve:

1. Compare the first two elements in the array, say A[1] and A[2]. If A[2] is less than A[1], then interchange the values.
2. Compare A[2] and A[3]. If A[3] is less than A[2], then interchange the values.
3. Continue this process till the last two elements are compared and interchanged.
4. Repeat the above steps for n-1 passes.

Points to remember:

- After the completion of first pass the largest element is placed in nth position.
- After second pass second largest element is placed in n-1th position.
- In the ith pass ith largest element is placed in n-(i-1)th position.
- If the elements are not in sorted order then it requires n-1 passes.
- If the element are in sorted order then it requires only one pass.