

INSERTION SORT -

The Insertion sort inserts each element in proper place. Suppose an array a with n elements $a[0], a[1], a[2], \dots, a[n]$ is in memory.

The insertion sort algorithm scans a from $a[0]$ to $a[n-1]$, inserting each element $a[k]$ into its proper position in the previously sorted subarray $a[0], a[1], a[2], \dots, a[k-1]$. That is.

Pass 1 $a[0]$ is trivially sorted.

Pass 2 $a[1]$ is inserted either before or after $a[0]$ so that $a[0], a[1]$ is sorted.

Pass 3

Pass N $a[n-1]$ is inserted in its proper place in $a[0], a[1], a[2], \dots, a[n-2]$ so that the array is sorted.

The element inserted in the proper place is compared with the previous elements and placed in between the i th element & $i+1$ th element if

element $\geq i$ th element.

element $\leq (i+1)$ th element

Let us take the elements -

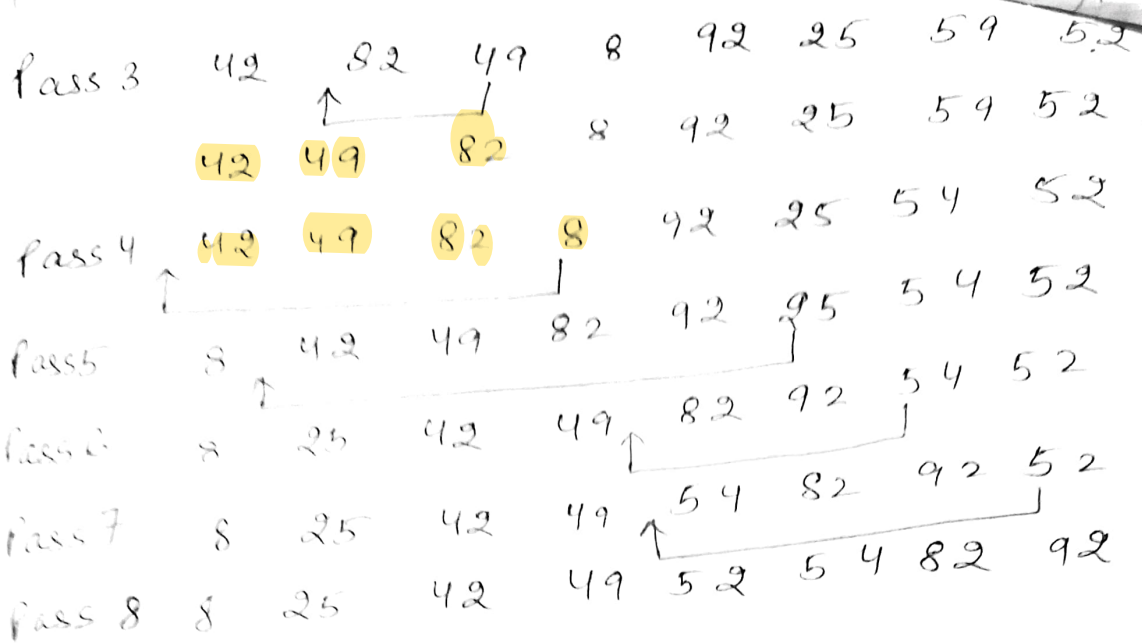
82 42 49 8 92 25 59 52

Pass 1 - 82 42 49 8 92 25 59 52

Pass 2 - 82 42 49 8 92 25 59 52

42

82



ALGORITHM - InsertionSort ($A[]$, Item)
 Let a be an array of n elements which we want to sort,

1. Set $K=1$
2. for $K=1$ to $(n-1)$
 Set $temp = a[K]$
 Set $j = K-1$
 While $temp < a[j]$ and $(j >= 0)$ perform the following steps.
 Set $a[j+1] = a[j]$
 [End of loop structure]
3. Exit

c. Function -

void insertion_sort ($int a[]$, $int n$)

{
 int $i, K, j, temp$;
 for ($K=1$; $K \leq n-1$; $K++$)
 { $temp = a[K]$;

$j = K-1$;
 while ($temp < a[j]$ & & $(j >= 0)$)
 {
 $a[j+1] = a[j]$;

$j = j - 1;$

$\{$
 $a[j+1] = temp;$

$\{$
printf ("Elements of array after sorting are: \n");

for (i

$\{$