

2. Insertion Sort :-

Insertion sort algorithm sorts a set of values by inserting value into an existing sorted file.

→ The element is inserted at a position where it gives sorted list.

→ That is, we start with the second element, compare it with the first and then the third is compared with the first two and so on.

→ Not suitable for large data sets.

Working :-

Pass 1 :- $A[1]$ by itself is trivially sorted.

Pass 2 :- $A[2]$ is inserted either before or after $A[1]$ so that: $A[1], A[2]$ is sorted.

Pass 3 :- $A[3]$ is inserted into its proper place in $A[1], A[2]$, that is, before $A[1]$, between $A[1]$ and $A[2]$, so that: $A[1], A[2], A[3]$ is sorted.

pass 4:- $A[4]$ is inserted into its proper place
in $A[1]$, $A[2]$, $A[3]$ so that:
 $A[1]$, $A[2]$, $A[3]$, $A[4]$ is sorted.

pass N : $A[N]$ is inserted into its
proper place in $A[1]$, $A[2]$... $A[N-1]$
so that: $A[1]$, $A[2]$... $A[N]$ is sorted.

* Algorithm (Insertion Sort)

Step 1:- If it is the first element, it
is already sorted.
return 1;

Step 2:- pick next element.

Step 3:- Compare with all elements in the sorted
sub list.

Step 4:- Shift all the elements in the sorted
sub list that is greater than the value
to be sorted.

Step 5:- insert the value.

Step 6:- Repeat until list is sorted.

123 971 80 7 92 20 UNSORTED NUMBERS

FIRST ITERATION

TEMP = 971

123 971 80 7 92 80

Compare 971 with

123 971 80 7 92 80

123, $123 < 971 < 123$

false so no operation

SECOND ITERATION

TEMP = 80

123 971 80 7 92 20

Compare 80 with 971, 80

123 ? 971 7 92 20

< 971 971 insert to 3rd p.

?

Compare 80 with 123, $80 < 123$

? 123 971 7 92 20

123 inserted 2nd position

80 123 971 7 92 20

80 inserted to 1st position

THIRD ITERATION

TEMP = 7

80 123 ? 971 92 20

Compare 7 with 971, $7 < 971$

971 inserted at 4th position

80 ? 123 971 92 20

Compare 7 with 123, $7 < 123$

123 inserted at 3rd position

? 80 123 971 92 20

Compare 7 with 80, $7 < 80$

80 inserted at 2nd position

7 80 123 971 92 20

7 inserted at 1st

position

