[Lect-26]

Insertion Sort :-

Say we have an array A with n elements.

A[1], A[2], A[3] - - - - A[N]

The insertion sort, insert each element A[K] into its proper position in the previously sorted subarray A[1], A[2] - - - -, A[K-1] such as

Pass1 A[1] by itself is trivially sorted

Pass2 A[2] is inserted either before or after A[1] so that: A[1], A[2] is sorted

Pass3 A[3] is inserted into its proper place in A[1], A[2] i.e before A[1], between A[1] & A[2] or After A[2] so that; A[1], A[2], A[3] is sorted

⋮

PassN 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.

- This algorithm is frequently used when n is small.
- Popular with bridge players when they are first sorting their Cards.

- How to insert $A[k]$ into its proper place in the sorted subarray. $A[1], A[2] - - - A[k-1]$

- We can do it by comparing $A[k]$ with $A[k-1]$, then with $A[k-2]$, then with $A(k-3)$ & so on.

- Until we meet an element $A[J]$ such that $A[J] \leq A[k]$

- This condition can be accomplished by introducing a sentinel element ie $A[0] = -\infty$ or a very small no. or we can have a condition in our loop

Taking example on Next page

An array A contains 8 element given below
77, 33, 44, 11, 88, 22, 66, 55

Pass	A[0]	A[1]	A[2]	A[3]	A[4]	A[5]	A[6]	A[7]	A[8]
k=1	$-\infty$	77	33	44	11	88	22	66	55
k=2	$-\infty$	77	33	44	11	88	22	66	55
k=3	$-\infty$	33	77	44	11	88	22	66	55
k=4	$-\infty$	33	44	77	11	88	22	66	55
k=5	$-\infty$	11	33	44	77	88	22	66	55
k=6	$-\infty$	11	33	44	77	88	22	66	55
k=7	$-\infty$	11	22	33	44	77	88	66	55
k=8	$-\infty$	11	22	33	44	66	77	88	55
Sorted list	$-\infty$	11	22	33	44	55	66	77	88

Procedure for insertion sort

```
Insertion ( A, N )
{
1.  for ( j = 2 to N )        // outer loop for passes
    {
2.      key = A[j]            // temporary variable to hold value.

3.      i = j-1               // Inserting A[j] into sorted segment A[1,2 ---- j-1]

4.      while ( i>0 && A[i] > key )
        {
            A[i+1] = A[i]
            i = i-1
        }                     // end of inner loop

5.      A[i+1] = key
    }                         // end of outer loop.
```

eg

9	6	5	0	8	2	7	1	3
1	2	3	4	5	6	7	8	9

assume it is itself sorted

j (so loop starts from index 2)

so key = 6 as j = 2

and i = 1 now (1 > 0 && 9 > 6) true

so we move the elements.

Complexity of Insertion Sort

~~No. of Comparison~~ ~~No. of~~

~~When j = 2~~

Complexity of Insertion = No. of Comparison + No. of movement

Pass	No. of Comparison	+	No. of movement	=	total operation
if $j = 2$	1	+	1	=	$2 \Rightarrow 2(1)$
$j = 3$	2	+	2	=	$4 \Rightarrow 2(2)$
$j = 4$	3	+	3	=	$6 \Rightarrow 2(3)$
⋮	⋮		⋮		
$j = n$	$n-1$	+	$n-1$	=	$2(n-1)$

So we write the above series.

when $j=2$, $j=3$, $j=4$, …, $j=n$

$$2(1) + 2(2) + 2(3) + \cdots\cdots 2(n-1)$$

$$2(1 + 2 + 3 + \cdots\cdots n-1)$$

$$2\left(\frac{(n-1)(n)}{2}\right) \Rightarrow \underline{\underline{O(n^2)}}$$

So worst case is $\underline{\underline{O(n^2)}}$

Best case :- when the list is already sorted.

1	2	3	4	5	6	7	8

↑ j

	no. of comparison	+	No. of movement	⇒	Total operation
$j=2$	1	+	0	⇒	1
$j=3$	1	+	0	⇒	1
$j=4$	1	+	0	⇒	1
⋮					
$j=n$	$(n-1)$	+	0	⇒	$n-1$

$\Rightarrow 1 + 1 + 1 + \cdots\cdots n-1.$

$\Rightarrow 1(n-1) \Rightarrow \underline{O(n)}$

Average case $\approx \underline{O(n^2)}$.