Saturday, March 6, 2021 11:30 AM

Behaviour of the Algorithm

for example:- Insertion Sort

A[n];

```
for(i=1 ; i<n; ++i){
  key = A[i];
   j = i-1;
  while(j>=0 && A[j]>key){
    A[j+1] = A[j];
    --j;
  } // End while
  A[j+1] = key;
} // End for
```

Time

$T_1 = n$

$T_2 = n$

$T_3 = n$

$T_4 = \sum_{j=0}^{} T_j$

$T_5 = T_4$

$T_6 = T_4$

$T_7 = n$

The algorithm behaviour will differ based on the given data, it will give different time complixities for different data.

① Worst case

Assume $A = \{50, 40, 30, 20, 10\}$ (0 1 2 3 4)

i	j	key	
1	0	40	50, 50, 30, 20, 10
	-1		40, 50, 30, 20, 10 — 1
2	1	30	40, 50, 50, 20, 10
	0		40, 40, 50, 20, 10
	-1		30, 40, 50, 20, 10 — 2
3	2	20	30, 40, 50, 50, 10
	1		30, 40, 40, 50, 10
	0		30, 30, 40, 50, 10 — 3
	-1		20, 30, 40, 50, 10
4	3	10	20, 30, 40, 50, 50
	2		20, 30, 40, 40, 50

4	3	10	20, 30, 40, 50, 50	
	2		20, 30, 40, 40, 50	
	1		20, 30, 30, 40, 50	4
	0		20, 20, 30, 40, 50	
	-1		10, 20, 30, 40, 50	

$$\sum_{j=0}^{n} j = \frac{n(n+1)}{2} = O(n^2)$$

(2) Best case ; Assume $A = \{10, 20, 30, 40, 50\}$

i	j	key		
1	0	20	same	1
2	1	30	"	1
3	2	4	"	1
4	3	50	"	1

best case $O(n)$

(3) Average Case : $A = \{20, 50, 10, 30, 40\}$

i	j	key		
1	0	50	"	0
2	1	10	20, 50, 50, 30, 40	
	0		20, 20, 50, 30, 40	2
	-1		10, 20, 50, 30, 40	
3	2	30	10, 20, 50, 50, 40	
	1		10, 20, 30, 50, 40	1
4	3	40	10, 20, 30, 50, 50	
	2		10, 20, 30, 40, 50	1

in Avg. $O(n^2)$

Search Algo. Best : Target element is the first

Avg. : = = in the middle

worst : = = is the last one.