· Analysis Of Algorithm

Psuedo Code:-

```
Insertion_sort (A):

1   for J=2 to A·length
2       Key = A[J]

3       // insert A[J] into the sorted sequence
             A[1·····J-1]

4       i = J-1

5       While  i>0  and  A[i] > Key
                A[i+1] = A[i]
6
7               i = i-1

8       A[i+1] = Key
```

· Dry Running:-

· WORST CASE:-

A = [4, 3, 2, 1]

1ST ITERATION OF FOR:-

1· J = 2

2· Key = 3

4· i = 1

1st iteration of while

```
5.  While 1>0 and 4>3
6.      A[2]=4 → A=[4,4,2,1]
7.      i=0
```

2nd iteration of while

```
5.  While 0>0
6.  ______
7.  ______

8.  A[1]=3 → A=[3,4,2,1]
```

• 2ND ITERATION OF FOR:-

```
1.  I=3
2.  Key=2
4.  i=2
```

1st iteration of while

```
5.  While 2>0 and 4>2
6.      A[3]=4 → A=[3,4,4,1]
7.      i=1
```

2nd iteration of while

```
5.  While 1>0 and 3>2
6.      A[2]=3 → A=[3,3,4,1]
7.      i=0
```

3rd iteration of while

```
5.  While 0>0
6.  ______
7.  ______

8.  A[1]=2 → A=[2,3,4,1]
```

• 3RD ITERATION OF FOR:-

1. J=4
2. Key=1
4. i=3

1st iteration of while

5. While 3>0 and 4>1
6. A[4]=4 → A=[2,3,4,4]
7. i=2

2nd iteration of while

5. While 2>0 and 3>1
6. A[3]=3 → A=[2,3,3,4]
7. i=1

3rd iteration of while

5. While 1>0 and 2>1
6. A[2]=2 → A=[2,2,3,4]
7. i=0

4th iteration of while

5. While 0>0
6. ______
7. ______

8. A[1]=1 → A=[1,2,3,4]

• 4TH ITERATION OF FOR:-

1. J=5

Line no	Time/ instruction	Frequency (worst case)	(best case)
1	C	n	n
2	C	n-1	n-1
3	0	n-1	n-1
4	C	n-1	n-1
5	C	$\sum_{J=2}^{n} J = \left[\frac{n(n+1)}{2} - 1\right]$	n-1
6	C	$\sum_{J=2}^{n} J-1 = \frac{n(n-1)}{2}$	0
7	C	$\sum_{J=2}^{n} J-1 = \frac{n(n-1)}{2}$	0
8	C	n-1	n-1

WORST CASE:-

$$T(n) = C\left[n + n-1 + n-1 + \left(\frac{n(n+1)}{2} - 1\right) + \frac{n(n-1)}{2} + \frac{n(n-1)}{2} + n-1\right]$$

$$T(n) = C\left[n+n+n+n-3 + \frac{n^2+n-2}{2} + \frac{n^2-n}{2} + \frac{n^2-n}{2}\right]$$

$$T(n) = C\left[4n-3 + \frac{n^2+n-2+n^2-n+n^2-n}{2}\right]$$

$$T(n) = C\left[\frac{8n-6+3n^2-n-2}{2}\right]$$

Teacher's Signature.

$$T(n) = c\left[\frac{3n^2 + 7n - 8}{2}\right]$$

$$T(n) = c[K_1 n^2 + K_2 n + K_3]$$

• DISCUSSION:-

- The worst case time complexity of insertion sort is $O(n^2)$
- Insertion sort in worst case grows quadratically
- Means that, maximum time which a code can take

• BEST CASE T(n):-

$$T(n) = [n + n-1 + n-1 + n-1 + n-1]c$$
$$T(n) = [5n-4]c$$

$$T(n) = [K_1 n + K_2]c$$

• DISCUSSION:-

- The best case time complexity of insertion sort is $O(n)$
- Insertion sort in best case grows linearly
- Means, that minimum time which a code can take.