

# Insertion Sort

$\checkmark$ 75,  $\checkmark$ 90,  $\checkmark$ 100,  $\checkmark$ 95,  $\checkmark$ 85,  $\checkmark$ 50,  $\checkmark$ 100,  $\checkmark$ 110,  $\checkmark$ 7  
 0 1 2 3 4 5 6 7 8

75	90
0	1

0 1 2  $90 < 100$   
 (Arrow from index 1 to index 0)

75	90	100
0	1	2

95  $100 > 95$   
 (Arrow from index 2 to index 1)

75	90	100	95	100
0	1	2	3	4

$90 > 85$   
 $95 > 85$   
 $85$  90  $95$   $100 > 85$   
 (Arrows from index 3 to index 2, then index 2 to index 1)

75	90	95	100	85	100
0	1	2	3	4	5

(Arrows from index 4 to index 3, then index 3 to index 2, then index 2 to index 1, then index 1 to index 0)

$75 > 50$   
 $85 > 50$   
 $90 > 50$   
 $95 > 50$   
 $100 > 50$

Go back until we locate the element that is less than the moving element

75	85	90	95	100	50
0	1	2	3	4	5

50 75 85 90 95 100  
 (Arrows from index 5 to index 4, then index 4 to index 3, then index 3 to index 2, then index 2 to index 1, then index 1 to index 0)

$100 < 110$

50	75	85	90	95	100	100	110	7
0	1	2	3	4	5	6	7	8

7 50 75 85 90 95 100 100 110  
 (Arrows from index 8 to index 7, then index 7 to index 6, then index 6 to index 5, then index 5 to index 4, then index 4 to index 3, then index 3 to index 2, then index 2 to index 1, then index 1 to index 0)

$110 > 7$   
 $100 > 7$   
 $100 > 7$   
 $95 > 7$   
 $90 > 7$   
 $85 > 7$   
 $75 > 7$   
 $50 > 7$

7, 50, 75, 85, 90, 95, 100, 100, 110

0 1 2 3 4 5 6 7 8

↳ sorted array

Best case

Ascending order  $\begin{cases} n-1 \text{ comparisons} \\ 0 \text{ swaps} \end{cases}$

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

$n=5$

10 ——— No comparison

10 | 20  
↻

$10 < 20$  ①

10 | 20 | 30  
↻

$20 < 30$  ②

10 | 20 | 30 | 40  
↻

$30 < 40$  ③

10 | 20 | 30 | 40 | 50  
↻

$40 < 50$  ④

$n$   $\begin{cases} (n-1) \text{ comparisons} = O(n) \\ 0 \text{ swaps} \end{cases}$

$$\underline{\underline{T(n) = O(n)}}$$

This is the Time complexity of Insertion sort in the best case scenario that is when the array is already sorted.

Please note that in case of sorting of an array Time complexity is measure of time taken for comparisons and time taken for swapping.

## Worst case

↳ Decending order

$$n = 5$$

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

n elements

C (Comparison)

S (Swaps)

50  
0

1

1

50 > 40

2

2

50 | 40 50  
0 1

3

3

50 > 30

4

4

40 > 30

1

1

40 | 50 | 30  
30 40 50

n-1

n-1

50 > 20

40 > 20

30 > 20

20 30 40 50  
30 | 40 | 50 | 20

↪ ↪ ↪

10 20 30 40 50

50 > 10

40 > 10

30 > 10

20 > 10

20 | 30 | 40 | 50 | 10  
↪ ↪ ↪ ↪

# comparisons → 1 + 2 + 3 + ... + n-1

$$\Rightarrow \frac{(n-1)n}{2} = O(n^2)$$

# Swaps

$$\hookrightarrow 1 + 2 + 3 + \dots + n-1$$

$$\frac{(n-1)n}{2} = O(n^2)$$

$$\underline{\underline{T(n) = O(n^2)}}$$

Worst case Time Complexity

Best case  $\rightarrow$  Ascending order

(Almost sorted  $\rightarrow$  ascending order)

**INSERTION  
SORT**  $\Rightarrow$

$\Downarrow$

$$\underline{O(n)}$$

Worst case  $\rightarrow$  Descending order

$\Downarrow$

$$\underline{O(n^2)}$$