

Insertion sort

0	1	2	3	4	5
<del>20</del>	<del>40</del>	<del>11</del>	<del>19</del>	<del>27</del>	45

n = 6

11 19 40 27 40 40

45 < 40

While (j > 0 and arr[j] < arr[j-1])

j = j - 1

swap(arr[j], arr[j-1])

j--

i = 2

j = ~~2~~ 1 0

i = 3

j = ~~3~~ 2 1

i = 4

j = ~~4~~ 3

Result

4 11, 19, 20, 27, 40, 45

(Ascending order)

i = 5

j = 5

Descending order

(worst case scenario)

~~40~~, ~~30~~, ~~20~~, ~~10~~, ~~5~~  
~~30~~ ~~40~~ ~~40~~ ~~40~~ ~~40~~  
~~20~~ ~~20~~ ~~10~~ ~~5~~  
~~10~~ ~~30~~ ~~5~~ ~~30~~  
~~5~~ ~~10~~ ~~20~~ ~~30~~  
~~5~~ ~~10~~

i = 1

j = 1

while (j > 0 && arr[j] < arr[j-1])

swap(arr[j], arr[j-1])

j--

comparision

1  
2  
3  
4  
|  
n-1

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

swaps

1  
2  
3  
4  
|  
n-1

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

Time complexity =  $O(n^2)$

Almost / fully  
sorted

10, 20, 40, 50, 8  
0 1 2 3 4  
5 15 5 5 50  
i=1 10 20 40 i=2 i=3  
J=1 5 < 50 J=2 J=3

while (j > 0 && arr[j] < arr[j-1])  
    swap(arr[j], arr[j-1])

J--;

<u>Comparison</u>	<u>Swaps</u>
1	0
1	0
1	0
1	0
1	0

Time complexity =  $O(n)$

Note:

1) Almost / fully sorted = Insertion sort  
     $O(n)$

2) Stable sort (Relative order is maintained)