

Sorting

4, 1, 5, 2, 3 \Rightarrow 1, 2, 3, 4, 5



Sorting

Bubble Sort

4, 5, 2, 3, 1
4, 2, 5, 3, 1
4, 2, 3, 5, 1
4, 2, 3, 1, 5

1st
pass

4, 2, 3, 1, 5
2, 4, 3, 1, 5
2, 3, 4, 1, 5
2, 3, 1, 4, 5

2nd
pass

2, 3, 1, 4, 5
2, 1, 3, 4, 5

3rd
pass

$j = 1 \rightarrow 2$

$j = 1 \rightarrow 4$

$j = 1 \rightarrow 3$

2, 1, 3, 4, 5
1, 2, 3, 4, 5

4th
pass

n sized array

n - 1 passes

Complete array is
sorted

$i = 1 \quad j = 2$

`int a = arr[i]`

0	1	2	3	4
1	2	3	4	5

$\quad \quad \quad \quad \quad$
 $\quad \quad \quad \quad \quad$
 $\quad \quad \quad \quad \quad$

`arr[i] = arr[j];` ←

$a = 2$

`arr[j] = arr[i]` ←
 $\quad \quad \quad \quad \quad$
 $\quad \quad \quad \quad \quad$

`for (int pass = 1; pass <= n-1; pass++)`
`{`

`for (int j = 1; j <= arr.length - pass; j++)`

`if (arr[j] < arr[j-1]) {`

`swap(j, j-1)`

`}`

`}`

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

$1 - n-1$

$$\frac{n \times (n-1)}{2}$$

$$\frac{n^2}{2} - \frac{n}{2}$$

$$\Rightarrow O(n^2)$$

Input $\rightarrow O(n)$

auxiliary $\rightarrow O(1)$

best case $\rightarrow \Omega(n)$

Selection Sort

0 1 2 3 4
4 1 5 2 3
1 4 5 2 3
1st pass

1 4 5 2 3
pos
1 2 5 4 3
2nd pass

0 1 2 3 4
1 2 5 4 3
pos
1 2 3 4 5
3rd pass

1 2 3 4 5
0 1 2 4
pos
4th pass

min = ~~4~~ / ~~1~~ / ~~3~~ / ~~2~~ / ~~5~~ / 3

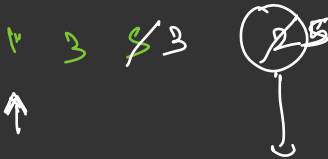
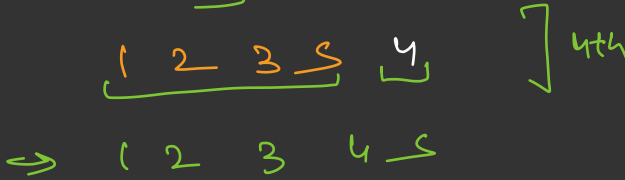
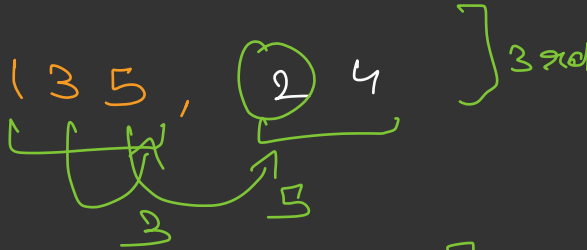
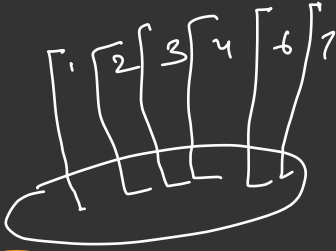
swap (pos, minIdx)

$L \rightarrow n$

passes = $n - 1 \Rightarrow TC = O(n^2)$

Auxiliary space $\Rightarrow O(1)$

Insertion Sort

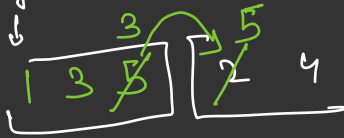


key = 2

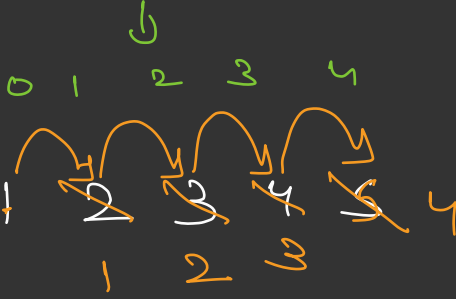


key = ~~3~~ 5 1 3 5 2 4

arr[j+1] = key;



key = 2



k = arr[l-1]

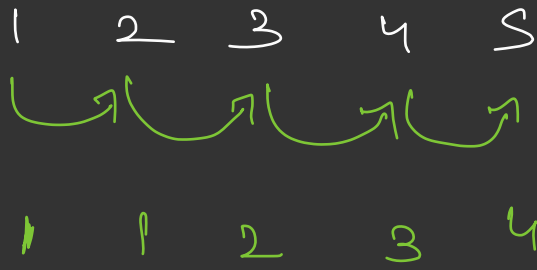
k = 5

while (j >= 0) {
 arr[j+1] = arr[j];
 j--;

}

→ previous last

shifting → end j >= 0



Time Complexity

4 5 3 2 1

$$1 + 2 + 3 + 4 + \dots + n-1$$

$$\rightarrow \frac{n \times (n-1)}{2} \Rightarrow \frac{n^2}{2} - \frac{n}{2} \quad O(n^2)$$

1 2 3 4 5

$$\underbrace{1 + 1 + 1 + \dots + 1}_{n-1} \Rightarrow \Omega(n)$$

inplace sorting

→ Array is sorted in input array

Space → Auxiliary space $\Rightarrow O(1)$

Ques Merge 2 sorted arrays

1	3	7	10	12
---	---	---	----	----

2	4	5	8	11	13	15
---	---	---	---	----	----	----

res

⇒

1	2	3	4	5	7	8	10	11	12	13	15
---	---	---	---	---	---	---	----	----	----	----	----

① Create new array ($n_1 + n_2$)
copy both the arrays ($n_1 + n_2$)
Sort the resultant array $n \log n$
 $n = n_1 + n_2$
Time ⇒ $n + n \log n \Rightarrow O(n \log n)$

②

1	3	7	10	12
---	---	---	----	----

↓ i

2	4	5	8	11	13	15
---	---	---	---	----	----	----

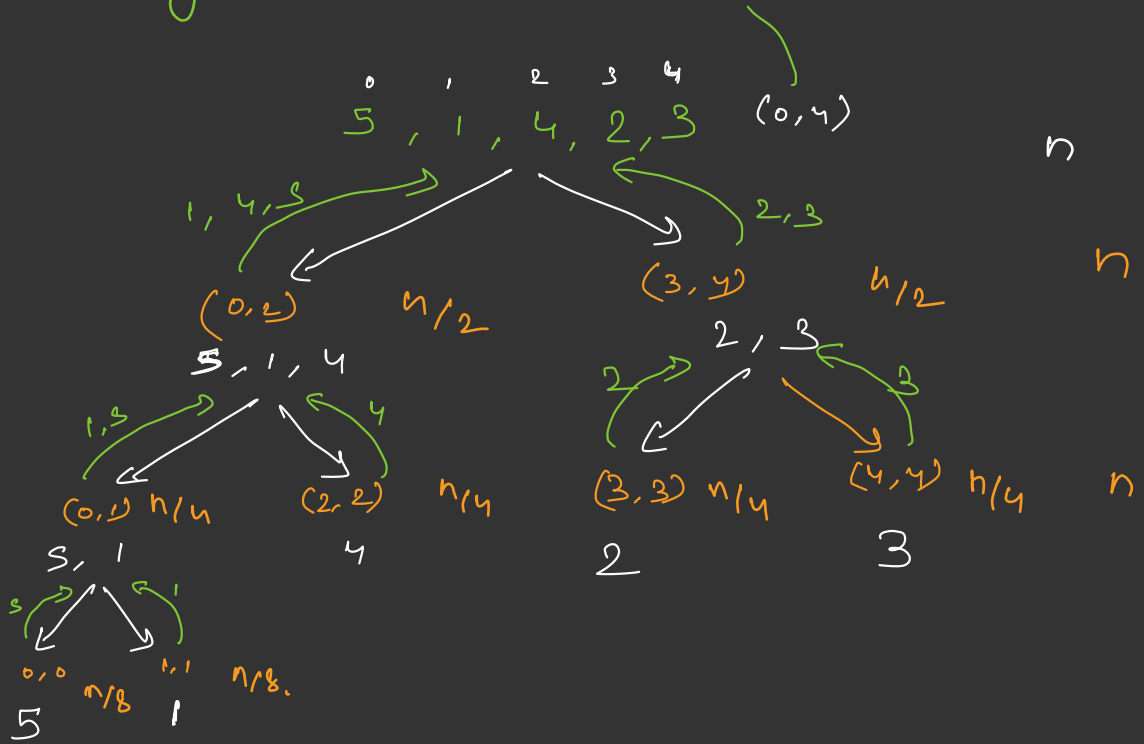
↓ j

1	2	3	4	5	7	8	10	11	12	13	15
---	---	---	---	---	---	---	----	----	----	----	----

↓ k

Merge Sort

1, 2, 3, 4, 5



$$\frac{n}{2^k} = 1$$

$$\Rightarrow k \Rightarrow \log_2(n)$$

$k n$ $\underbrace{n + n + n \dots}_{k \text{ step}}$

Time $\Rightarrow O(n \log n)$

not inplace

Space \rightarrow Auxiliary Space $\Rightarrow O(n)$

