

## Sorting

↳ Arrange list of numbers  
in asc or desc order

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

## Bubble Sort

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

1st Pass

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

$i = 1, n-1$

$i = 1, n-2$

1, 2, 3, 4

n sized array

$\Rightarrow$

n-1 passes

Completely  
sort array.

1, 2, 3, 4, 5

isSwapped = false

for (int pass = 1; pass < a.l; pass++) {

for (i = 1; i <= a.l - pass; i++) {

if (arr[i-1] > arr[i])  
swap (arr, i, i-1) }  
isSwapped = true

3  
3

1 (n-1) x k  
2 (n-2) x k  
3 (n-3) x k  
⋮  
n-1 1 x k

$$\frac{(1 + 2 + 3 + \dots + n-1) \times k}{2} = O(n^2)$$

TC  $\Rightarrow O(n^2)$

$\Omega(n)$

Input  $\Rightarrow O(n)$

Auxiliary space  $\Rightarrow O(1)$

$\rightarrow$  in place sorting  $\rightarrow$  sorted in input array  
stable

# Selection Sort

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

1 4 5 2 3  
↑  
pass  
→ 1 2 5 4 3  
2nd pass

1 2 3 4 5  
↑  
pass  
→ 1 2 3 4 5  
3rd pass

1 2 3 4 5  
↑  
pos  
4th pass

$\text{minIdx} = 1$

$\text{pos} = 2$

$\text{swap}(\text{pos}, \text{minIdx})$

passes  $\Rightarrow n-1$

$\text{pos} = \text{pass} - 1$

TC  $\Rightarrow O(n^2)$

Input  $\Rightarrow O(n)$


Auxiliary  $\Rightarrow O(1)$

In place sorting


Stable

# Bubble sort

5, 4, 3, 2, 1



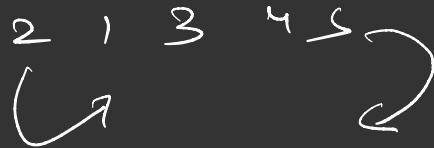
4, 3, 2, 1, 5



3, 2, 1, 4, 5

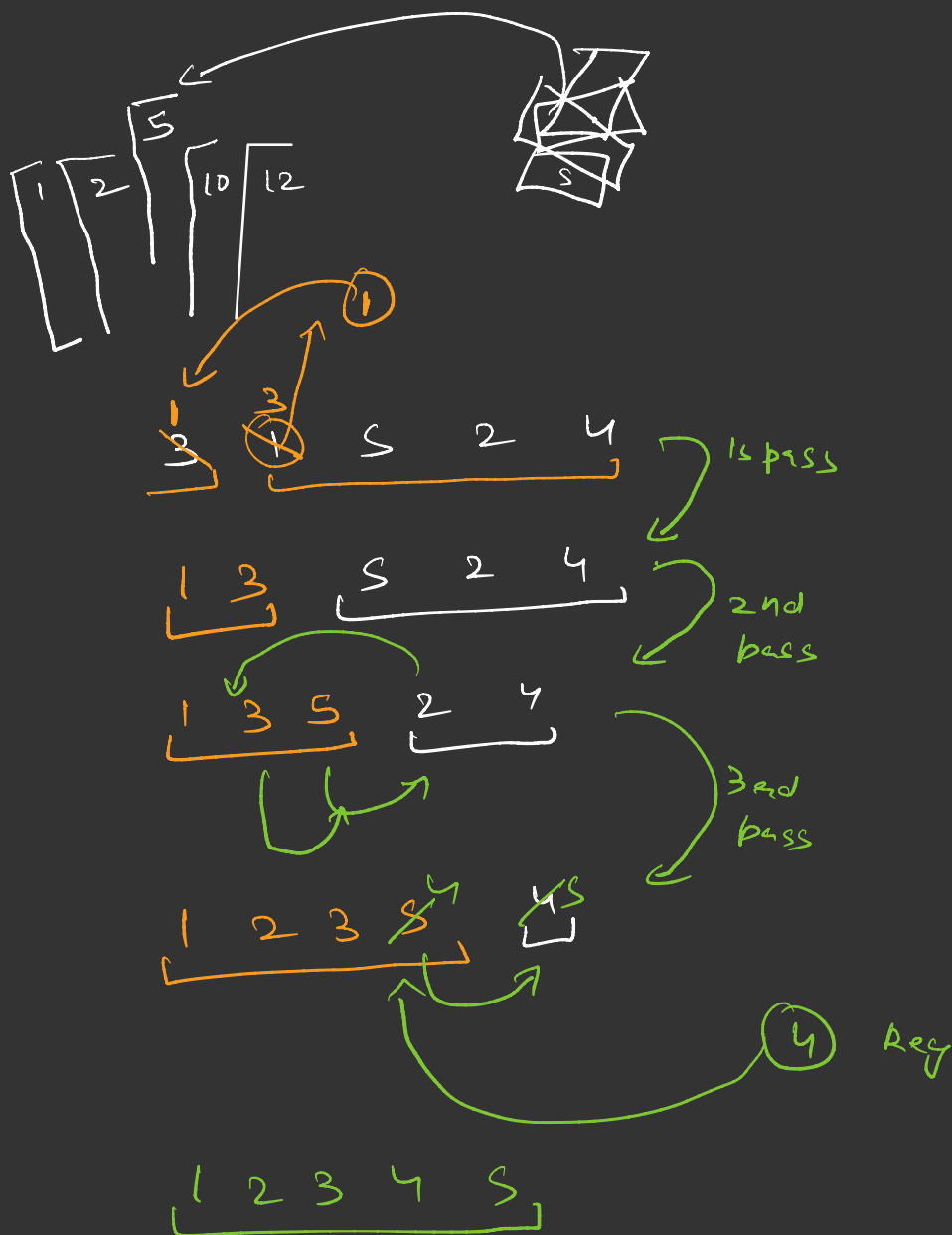


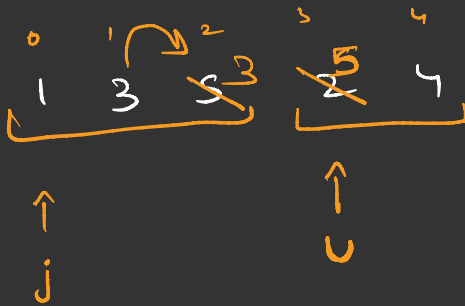
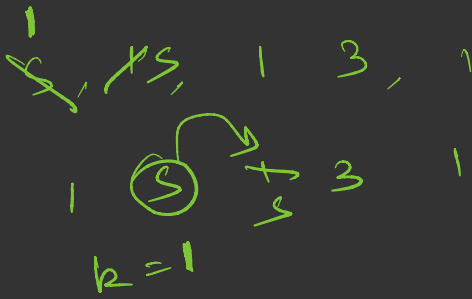
2, 1, 3, 4, 5



1, 2, 3, 4, 5

# Insertion Sort





k = 2

1 → 0

2 → 1

3 → 2

n → n-1

TC  $\Rightarrow O(n^2)$

Input  $\Rightarrow O(n)$

Auxiliary Space  $\Rightarrow O(1)$   
 inplace  
 stable