

# Sorting

↳ Ascending Order  
or      Descending Order

Ans: [10, 20, 40, 9, 4, 7] → not sorted

[10, 20, 40, 100, 120] → Sorted or unsorted

↳ Ascending Order ✓

[100, 40, 20, 6, -1, -4] → Sorted

↳ Descending order:

[100, 40, 6, 90, 120] → unsorted.

Ascending Order

## Algorithm → set of steps to solve a problem

Q. Find sum of elements of array

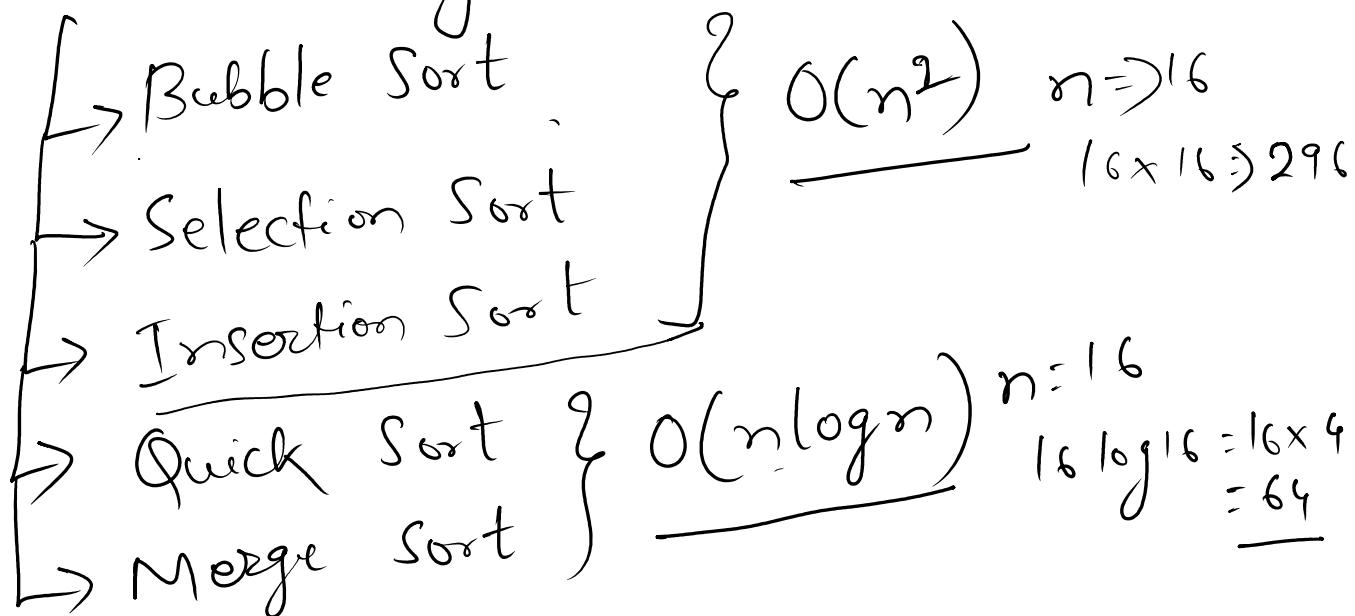
Step 1: - Initialize a variable with 0.

Step 2: - Iterate over entire array

Step 3: - Add each array element to variable

- Algorithm:
  - 1. Brute Force Algorithm.
  - 2. Predefined Algorithm.

Sorting: - To make elements of array in ascending order or descending order

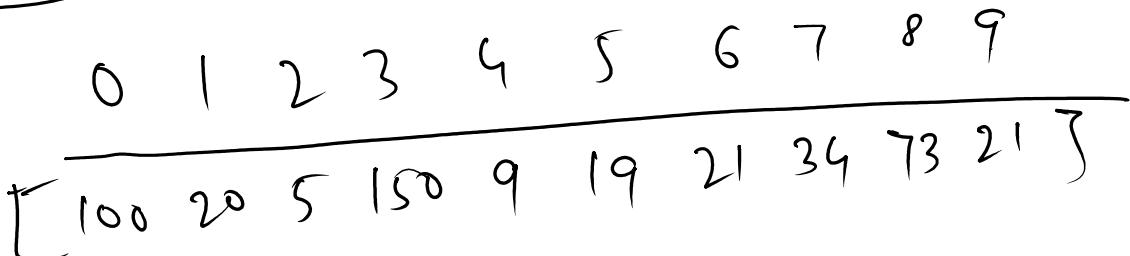


Data Structure :- Way to store data.

↳ Linear Data Structure

↳ Linear Data Structure  
↳ Non Linear Data Structure

### Linear Data Structure -



Example - (i.) Array  $\rightarrow [100, 50, 60, 1]$

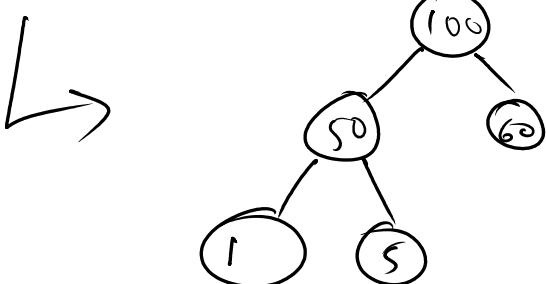
(ii.) LinkedList  $\rightarrow [100] \rightarrow [50] \rightarrow [60] \rightarrow [1]$

(iii.) Stack  $\rightarrow \begin{matrix} 60 \\ 50 \\ 1 \end{matrix}$

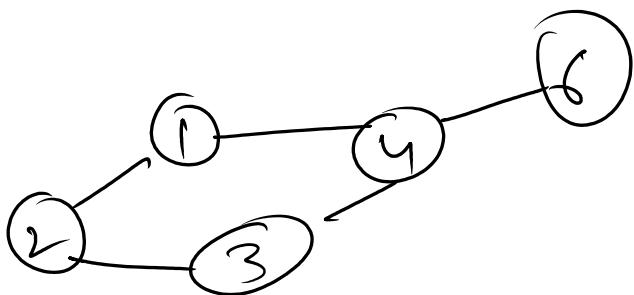
(iv.) Queue  $\rightarrow \begin{matrix} \text{Front} & 100, 50, 60, 1 & \text{Rear} \end{matrix}$

### Non Linear Data Structure

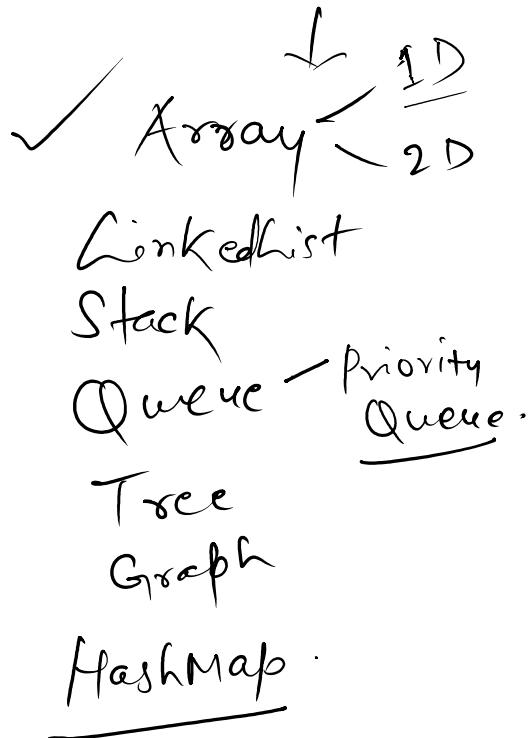
#### Tree



#### Graph



# DSA → Data Structure & Algorithm



Sorting Algorithm  
Two Pointer Algorithm  
Searching Algorithm  
Sliding Window Algorithm  
Kadane's Algorithm  
Dynamic Programming

Bubble Sort  $\rightarrow O(n^2)$

(Ascending order)

$arr[j] = [100, 10, 9, 200, 6]$

$i=0$

$j=0 \text{ to } n-2$   $j < n-1$

$j=0$   $[10, \underline{100}, 9, 200, 6]$

$j=1$   $[10, 9, \underline{100}, \underline{200}, 6]$

$j=2$   $[10, 9, 100, \underline{200}, 6]$

$j=3$   $[10, 9, 100, 6, \underline{\underline{200}}]$

$n=5$  (length)

$0 \ 1 \ 2 \ 3 \ 4$   
 $\underline{[6, 9, 10, 100, 200]}$

$i=1$   $5-1-1=3$   $0 \ 1 \ 2 \ 3 \ 4$   
 $j=0 \text{ to } 3$   $< n-i$   $[10, 9, 100, 6, 200]$

if ( $arr[j] > arr[j+1]$ ) {

swap ; }

$j=0 \ 0 \ 1 \ 2 \ 3 \ 4$   
 $\underline{[9, 10, 100, 6, 200]}$

$j=1$   
 $[9, 10, 100, 6, 200]$

$j=2$   
 $[9, 10, 6, \underline{100}, 200]$

$arr[j] > arr[j+1]$

swap

↓  
1 - 100

$[9, 10, 6, 100, 200]$  ↗ swap

↙ false,

$j = \underline{3}$  0 1 2 3 4  
 $[9, 10, 6, 100, \underline{200}]$

$i = 2$  0 1 2 3 4  
 $j = 0 \rightarrow 3$  0 1 2 3 4  
 $[9, 10, 6, 100, 200]$

$j = \underline{0}$  0 1 2 3 4  
 $[9, \underline{10}, 6, 100, 200]$

$j = \underline{1}$  0 1 2 3 4  
 $[9, 6, \underline{10}, 100, 200]$

$j = 2$  0 1 2 3 4  
 $[9, 6, 10, 100, 200]$

$j = 3$  0 1 2 3 4  
 $[9, 6, 10, \underline{100}, \underline{200}]$

$arr[j] > arr[j+1]$

$arr[2] > arr[3]$

↙ swap.

$i = 3$  0 1 2 3 4  
 $j = 0 \rightarrow 3$  0 1 2 3 4  
 $[9, 6, 10, 100, 200]$

$j = 0$  0 1 2 3 4  
 $[6, 9, 10, 100, 200]$

$j = 1$  0 1 2 3 4  
 $[6, 9, 10, 100, 200]$

$j = 2$  0 1 2 3 4  
 $[6, 9, 10, 100, 200]$

⑤ → ⑤

$i = 0 \rightarrow 4$   
 $0 \rightarrow n-1$

j=2 [6, 9, 10, 100, 200]

j=3 [6, 9, 10, 100, 200]

i=4.

j=0 [6, 9, 10, 100, 200]

j=1 to 3

n=5

1, 2, 3, 4, 5  
0, 1, 2, 3, 4

~~O(n<sup>2</sup>)~~ for (int i=0; i< n; i++) {  
~~O(n<sup>2</sup>)~~ for (int j=0; j< n-1; j++) {  
    → j < n-i-1     (Optimized)  
    0, 1, 2, 3, 4, ..., n-2

        if (arr[j] > arr[j+1]) {

            int temp = arr[j];

            arr[j] = arr[j+1];

            arr[j+1] = temp;

}

}

}

for (int i=0; i< n; i++) {

```
s.o.p(arry[i]+“ ”),  
{
```