Data Structures

Sorting Technique – Introduction

CODE FORTHINGS

Introduction

Introduction

Sorting:

It is a process of arranging items systematically

Data Structure – Sorting Techniques

Introduction

Sorting:

It is a process of arranging items systematically

Meaning:

Ordering or Arranging the items in a sequence ordered

Introduction

Sorting Algorithm:

Is an Algorithm that puts the elements of a list in a particular order

Introduction

Sorting Algorithm:

Is an Algorithm that puts the elements of a list in a particular order

Data Searching is optimized

Introduction

Sorting Techniques:

- Bubble Sort
- Insertion Sort
- Selection Sort
- Quick Sort
- Merge Sort
- Heap Sort

Data Structure – Sorting Techniques

Bubble Sort

- It is simple sorting technique

- This sorting algorithm is comparison-based algorithm in which each pair of adjacent elements is compared and the elements are swapped if they are not in order.

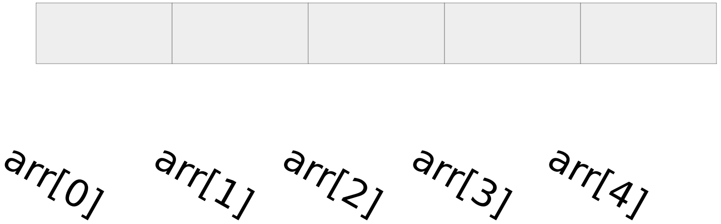
- Its average and worst case complexity are of $O(n^2)$ where n is the number of items.

Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5

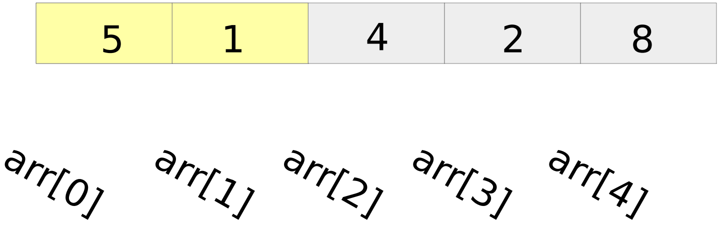


Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5



arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
5	1	4	2	8

5 > 1

Bubble Sort

- arr[SIZE]

SIZE = 5

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
5	1	4	2	8

5 > 1

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	5	4	2	8

5 > 4

Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5

5	1	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 1

1	5	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 4

1	4	5	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 2

Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
5	1	4	2	8

5 > 1

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	5	4	2	8

5 > 4

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	5	2	8

5 > 2

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	2	5	8

5 < 8

Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5

5	1	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 1

1	5	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 4

1	4	5	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 2

1	4	2	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 < 8

1	4	2	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

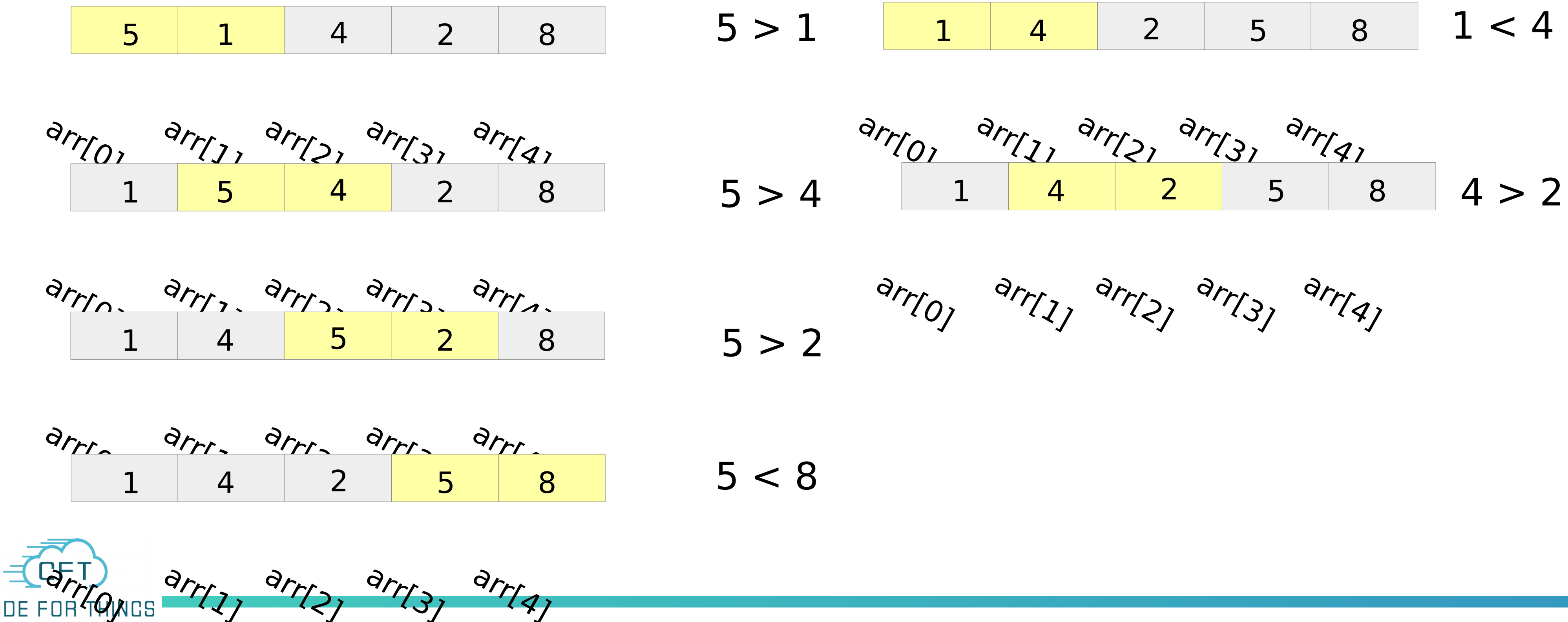
1 < 4

Data Structure – Sorting Techniques

Bubble Sort

- arr[SIZE]

SIZE = 5



arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
5	1	4	2	8

5 > 1

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	5	4	2	8

5 > 4

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	5	2	8

5 > 2

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	2	5	8

5 < 8

arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	2	5	8

1 < 4

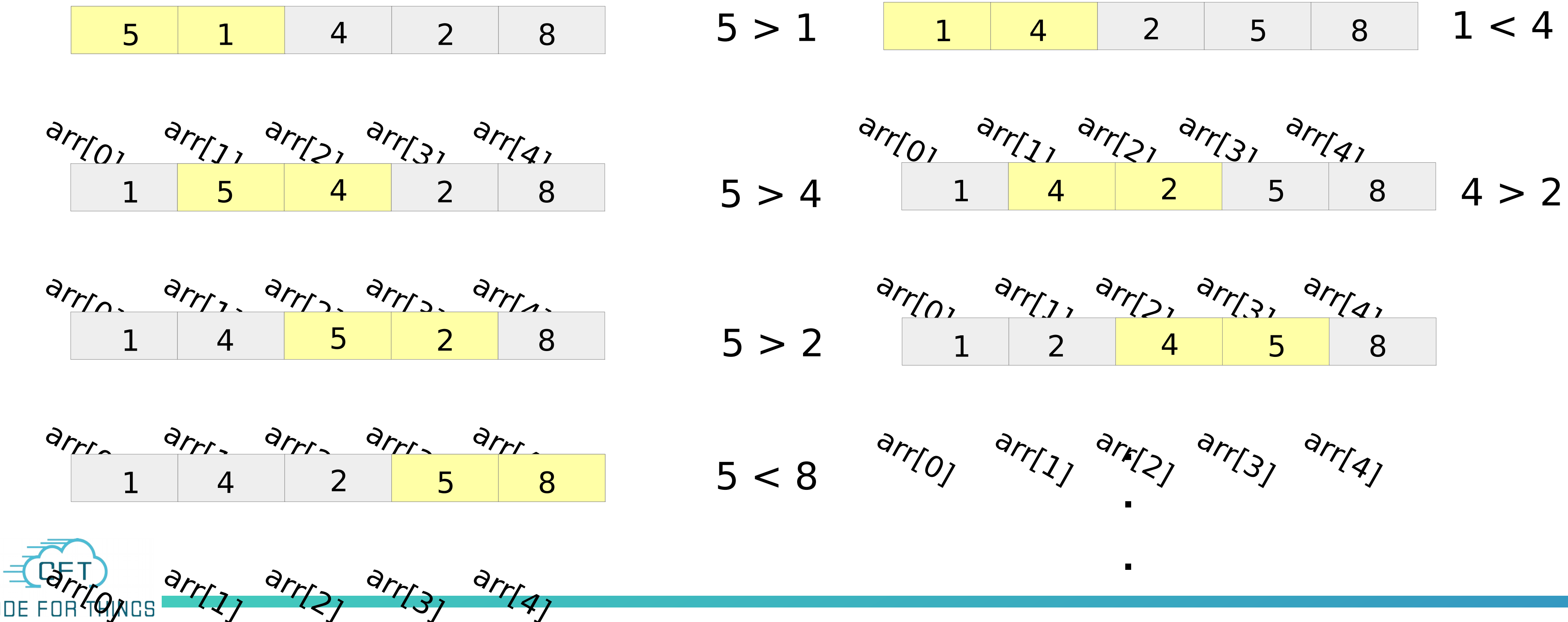
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]
1	4	2	5	8

4 > 2

Bubble Sort

- arr[SIZE]

SIZE = 5



5	1	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 1

1	5	4	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 4

1	4	5	2	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 > 2

1	4	2	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

5 < 8

1	4	2	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

1 < 4

1	4	2	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

4 > 2

1	2	4	5	8
arr[0]	arr[1]	arr[2]	arr[3]	arr[4]

.

.

.

Algorithm

Bubble Sort(arr,size)

Algorithm

Bubble Sort(arr,size)

```
for i = 0  upto  size
 for j = 0  upto size-1-i
        If (arr[j] > arr[j+1])
            temp = arr[j]
            arr[j] = arr[j+1]
            arr[j+1] = temp
return e_true
```

Bubble Sort

Advantages

- It is easy to implement
- Elements are swapped in place without using additional temporary storage.
- The space requirement is at a minimum

Bubble Sort

Advantages

- It is easy to implement
- Elements are swapped in place without using additional temporary storage.
- The space requirement is at a minimum

Disadvantages

- The bubble sort requires n-squared processing steps for every n number of elements to be sorted.
- Time complexity = $O(n^2)$

Code -Bubble Sort