Data Structures

Sorting Technique – Insertion Sort

CODE FORTHINGS

Introduction

Data Structure – Sorting Techniques

Introduction

Insertion Sort:

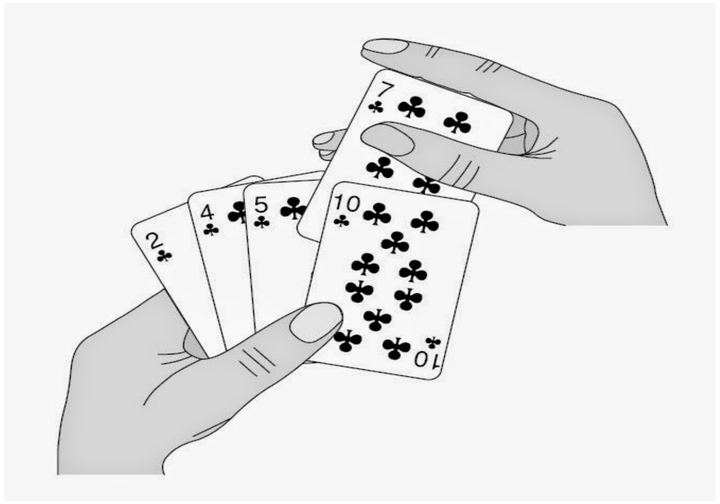
- It is a simple sorting algorithm that builds the final sorted array(or list) one item at a time

Data Structure – Sorting Techniques

Introduction

Insertion Sort:

- It is a simple sorting algorithm that builds the final sorted array(or list) one item at a time
- It is an in-place comparison-based algorithm

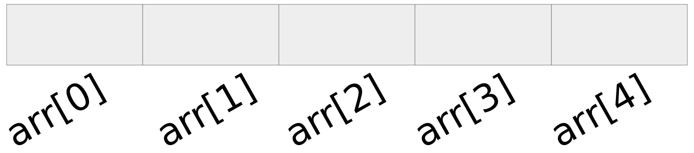


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

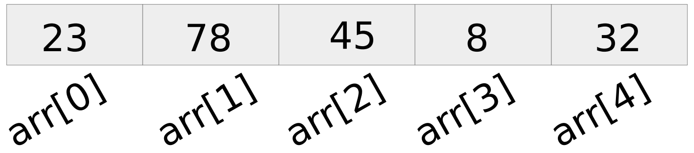


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

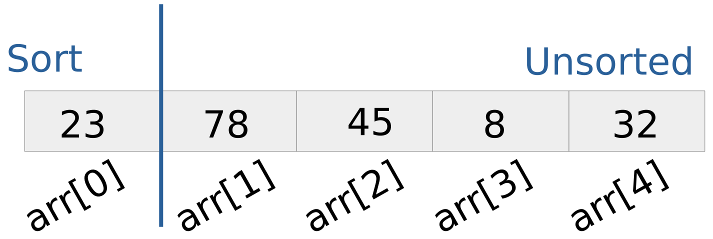


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

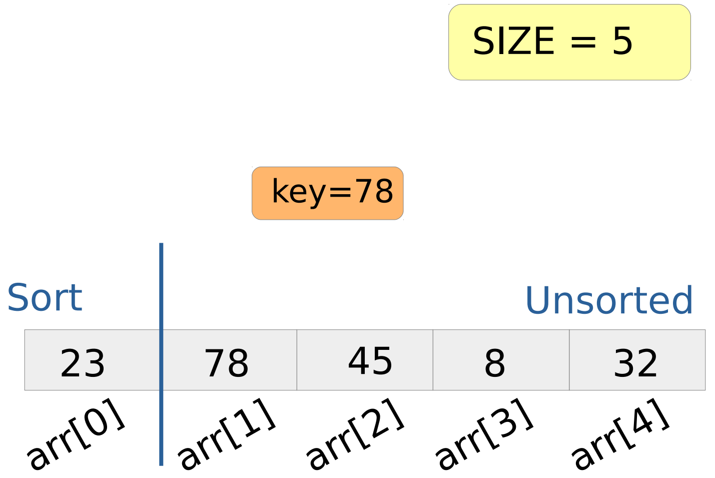
SIZE = 5



Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

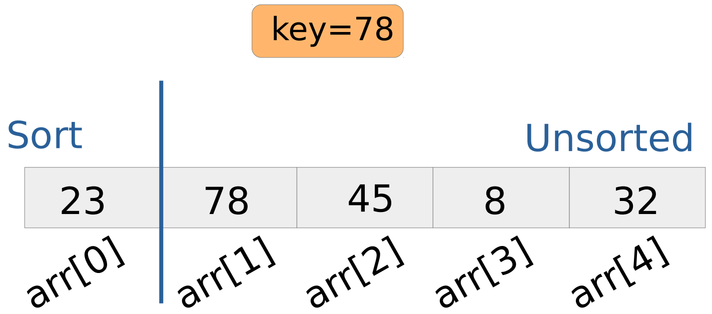


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[0]

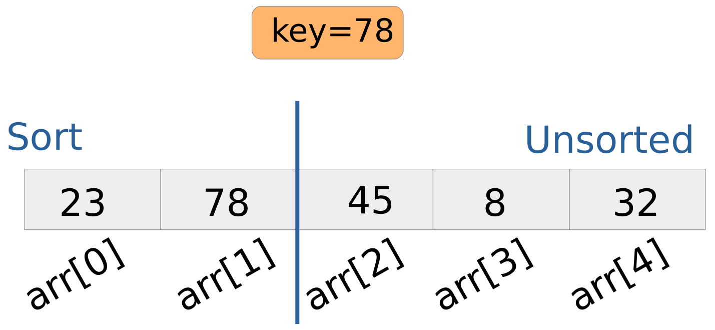
$78 \nless 23$

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



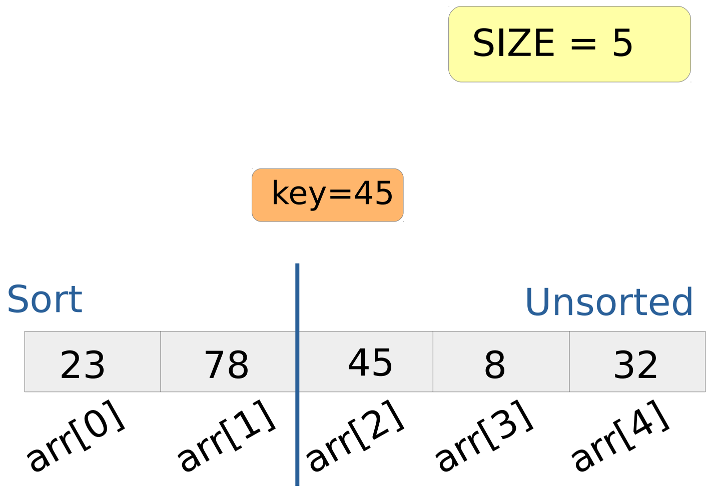
key < arr[0]

78 < 23

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]



key < arr[1]

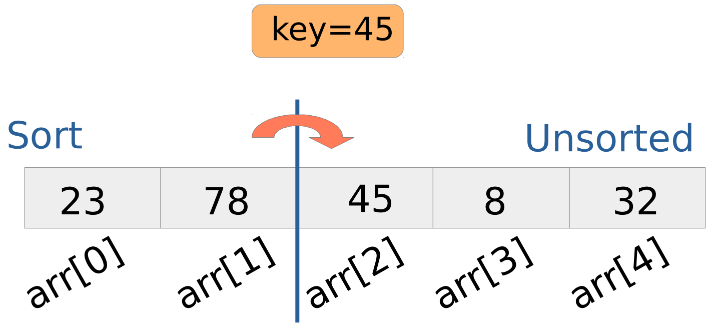
45 < 78

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[1]

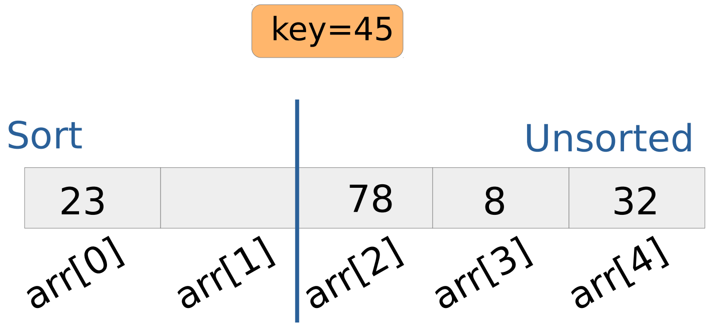
45 < 78

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[1]

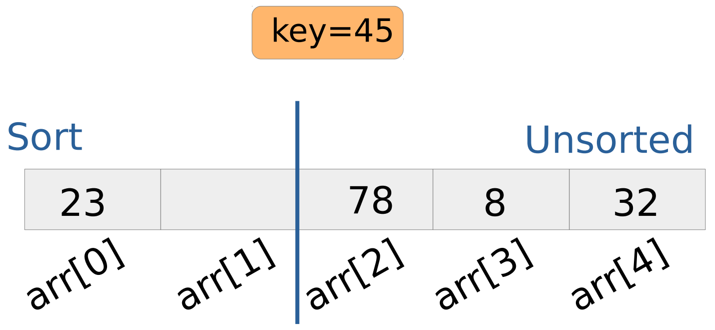
45 < 78

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[1]

45 < 78

key < arr[0]

$45 \not< 23$

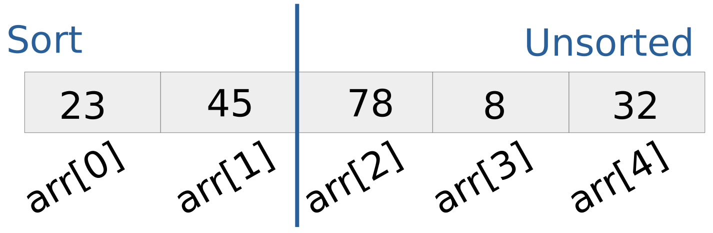
Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

key=45



key < arr[1]

45 < 78

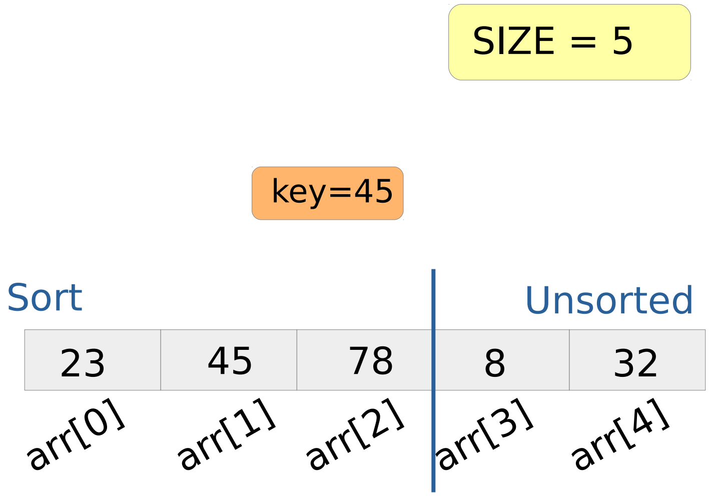
key < arr[0]

45 < 23

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

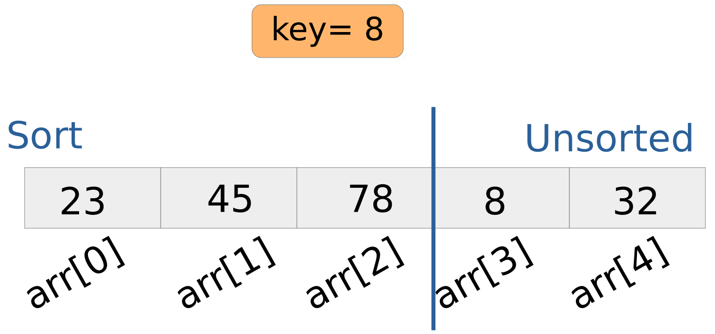


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[2]

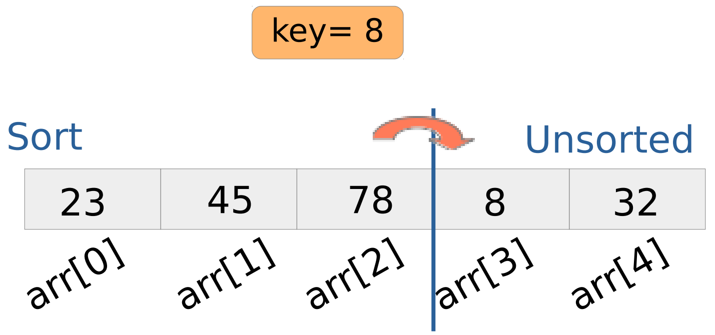
8 < 78

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[2]

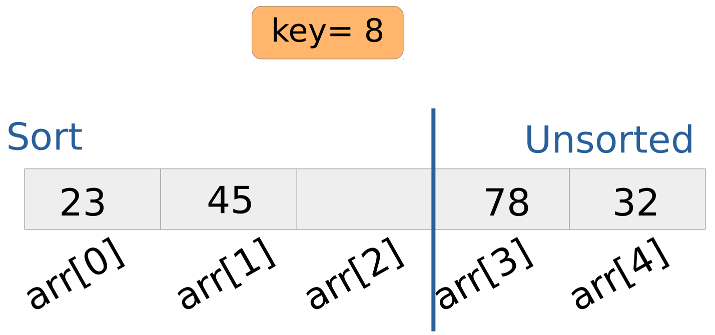
8 < 78

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[2] key < arr[1]

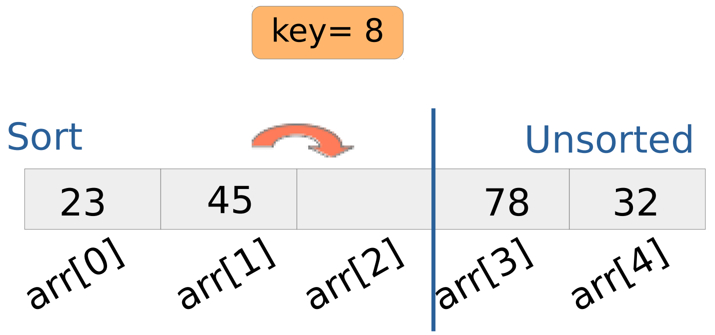
8 < 78 8 < 45

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[2]

8 < 78

key < arr[1]

8 < 45

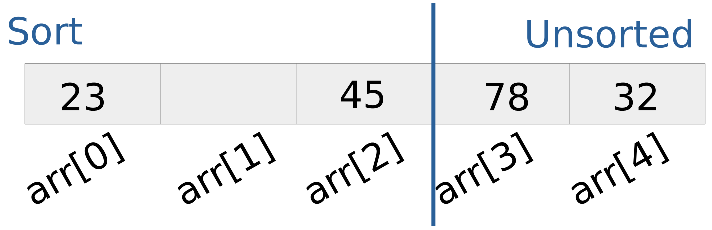
Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

key= 8



key < arr[2]

$8 < 78$

key < arr[1]

$8 < 45$

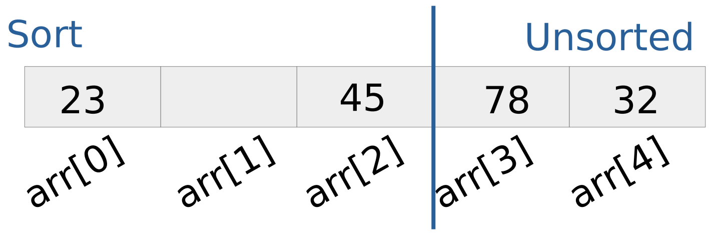
Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

key= 8



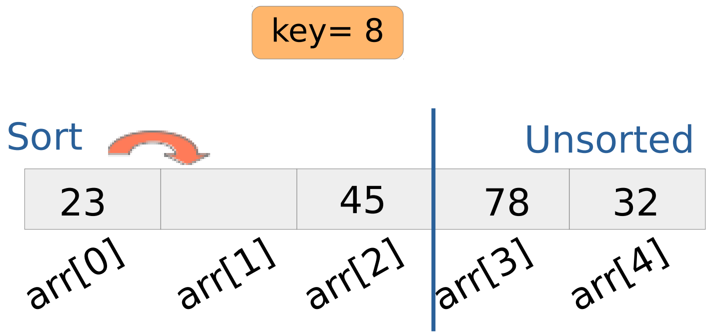
key < arr[2]	key < arr[1]	key < arr[0]
8 < 78	8 < 45	8 < 23

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



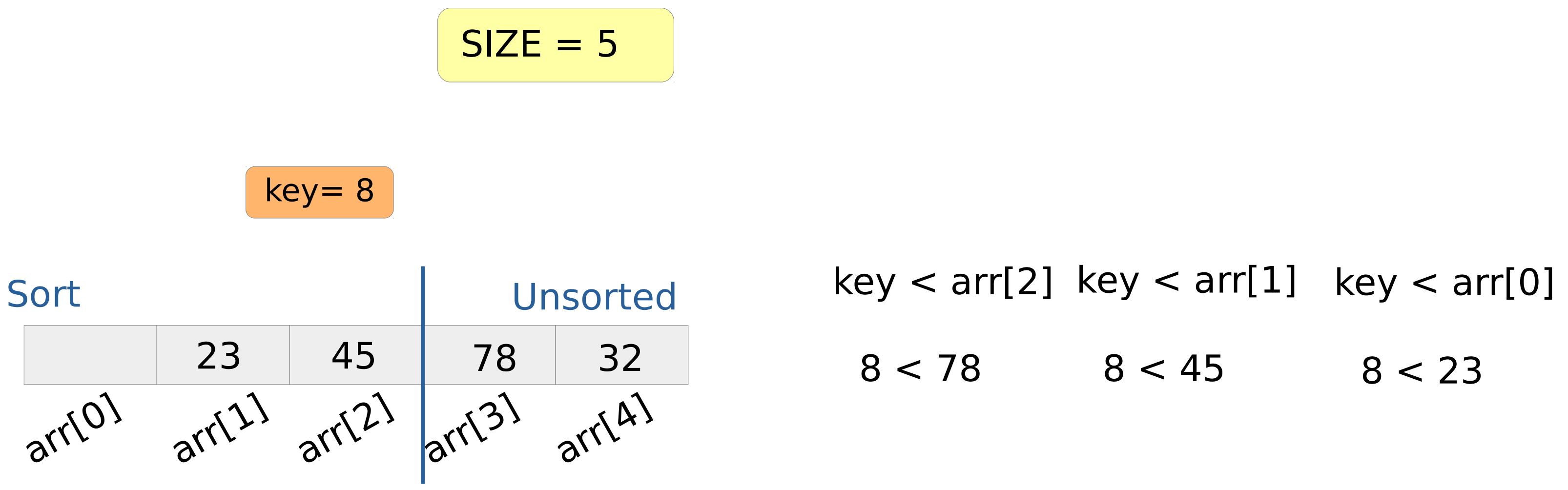
key < arr[2] key < arr[1] key < arr[0]

8 < 78 8 < 45 8 < 23

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]



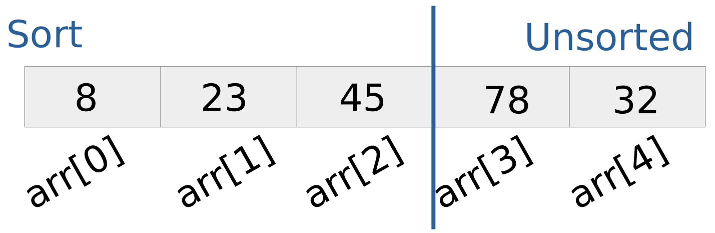
Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5

key= 8

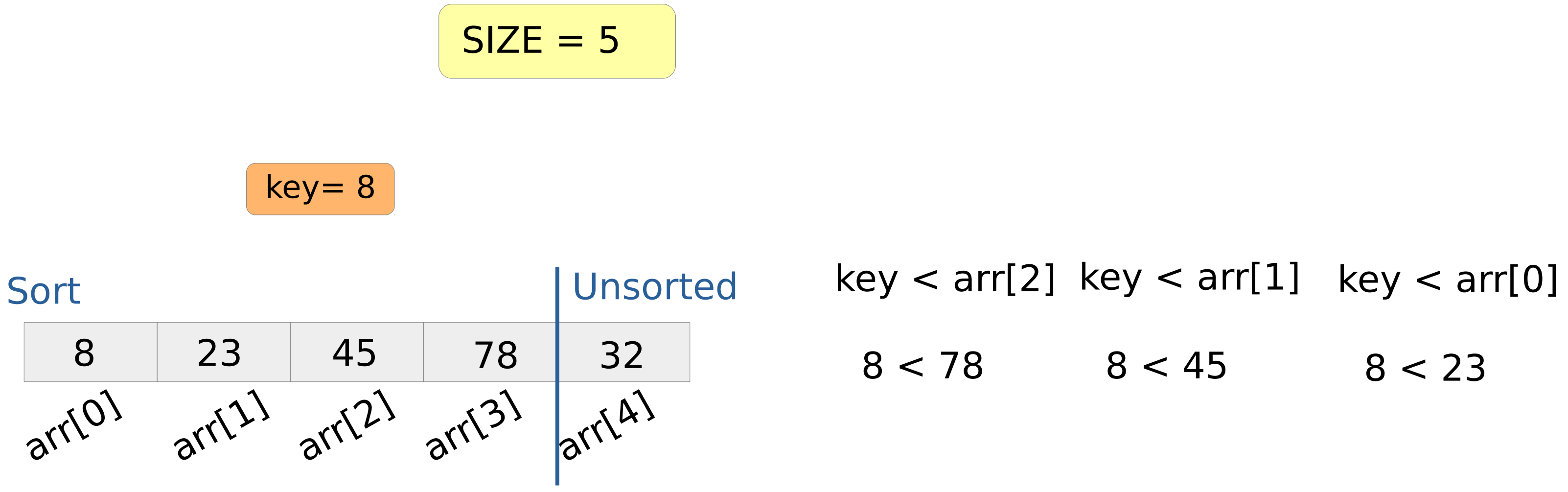


key < arr[2]	key < arr[1]	key < arr[0]
8 < 78	8 < 45	8 < 23

Data Structure – Sorting Techniques

Insertion Sort

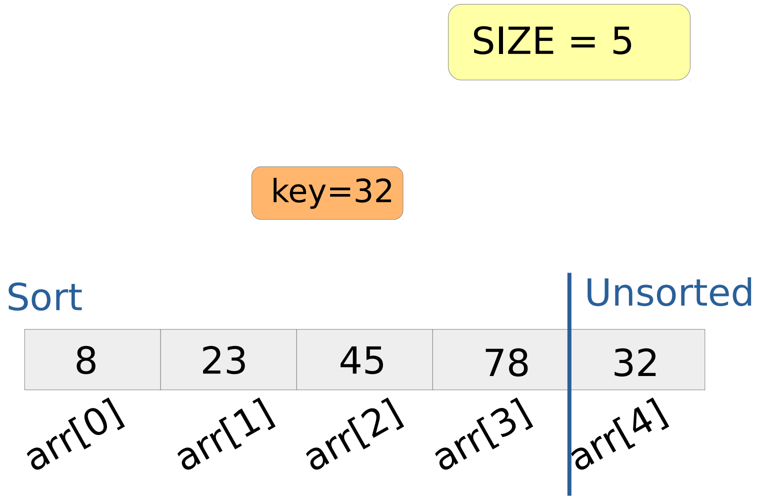
- arr[SIZE]



Data Structure – Sorting Techniques

Insertion Sort

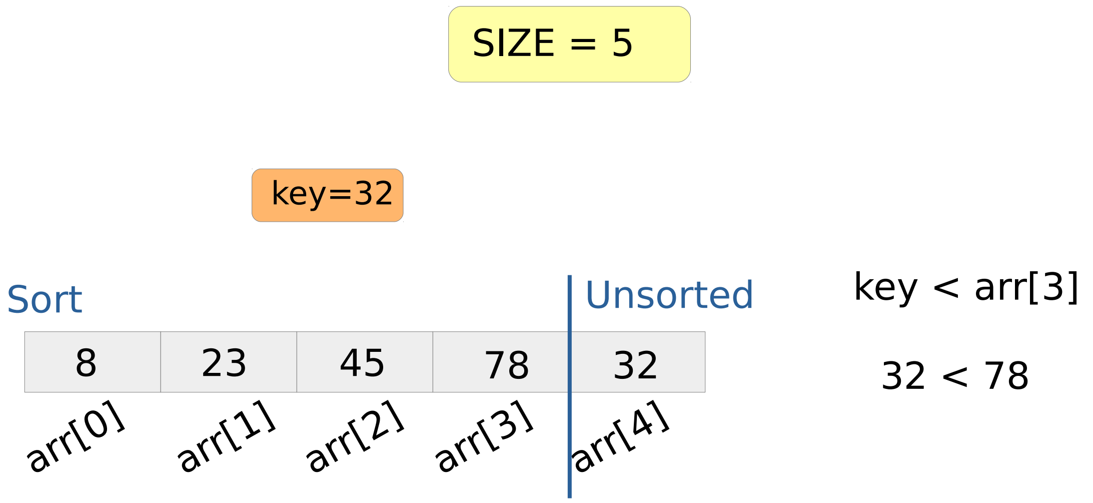
- arr[SIZE]



Data Structure – Sorting Techniques

Insertion Sort

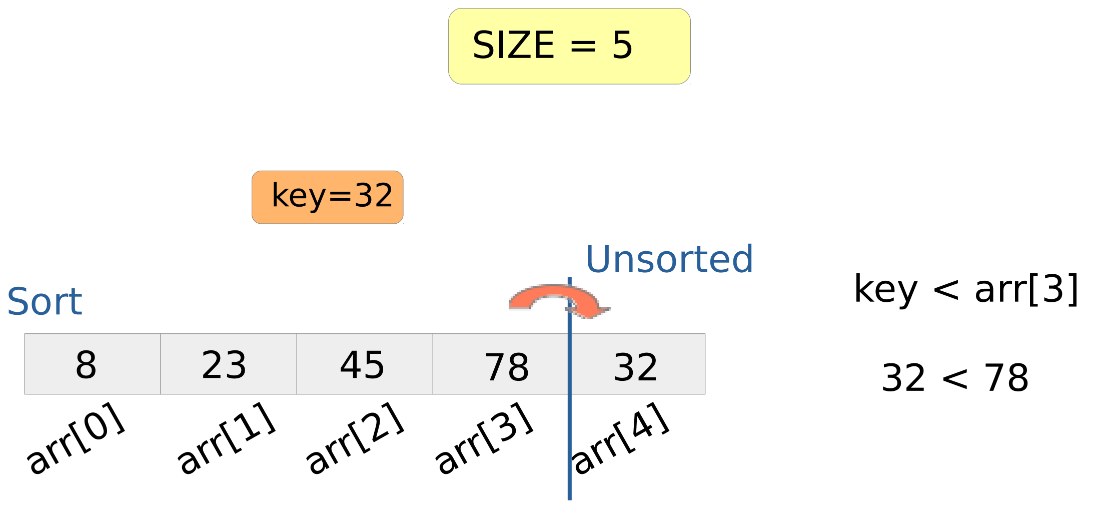
- arr[SIZE]



Data Structure – Sorting Techniques

Insertion Sort

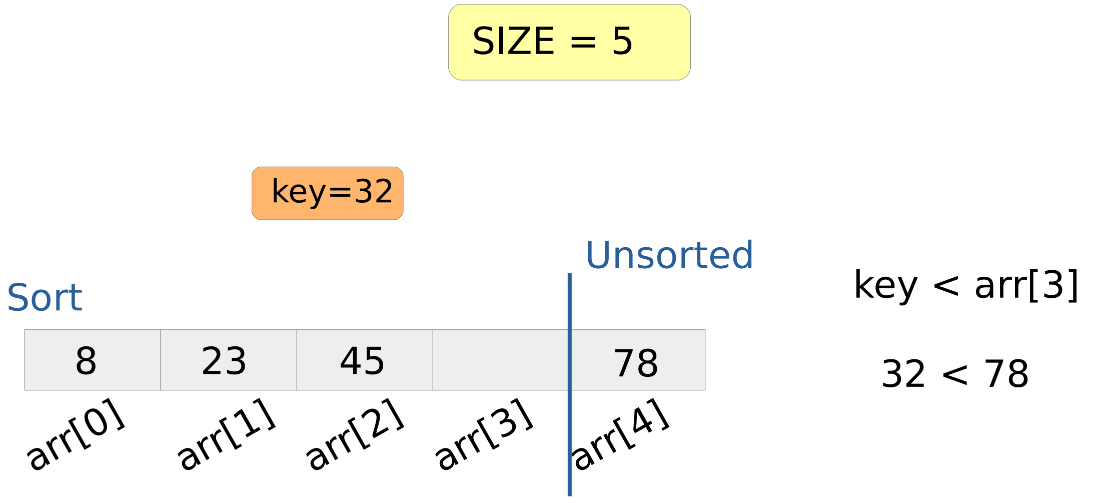
- arr[SIZE]



Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

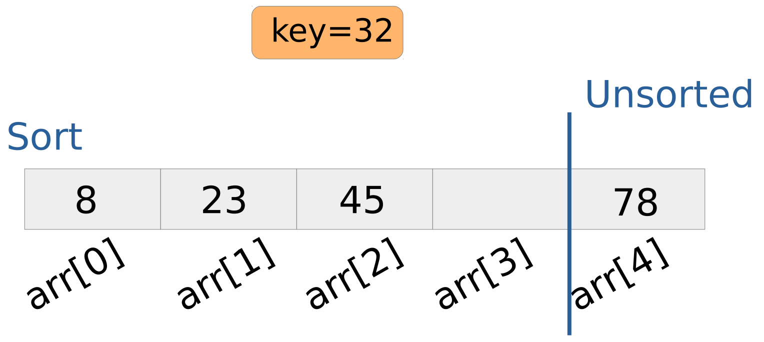


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



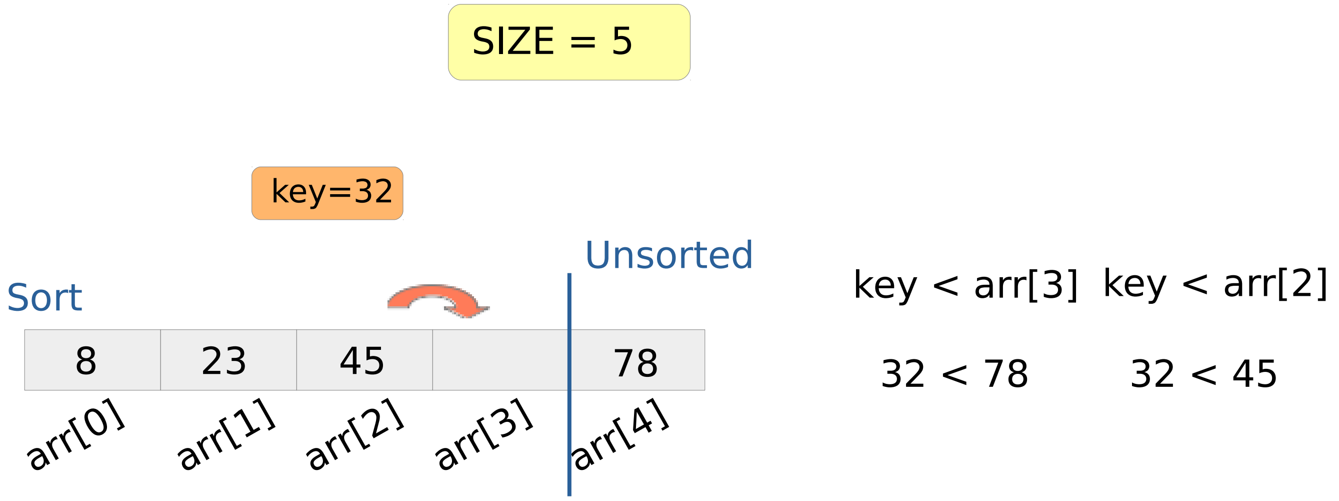
key < arr[3]   key < arr[2]

32 < 78   32 < 45

Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

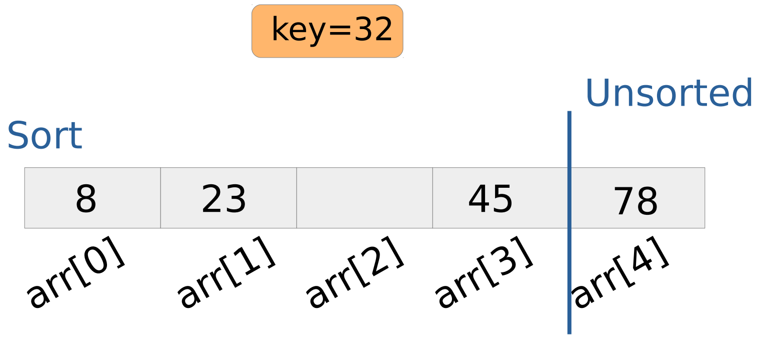


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



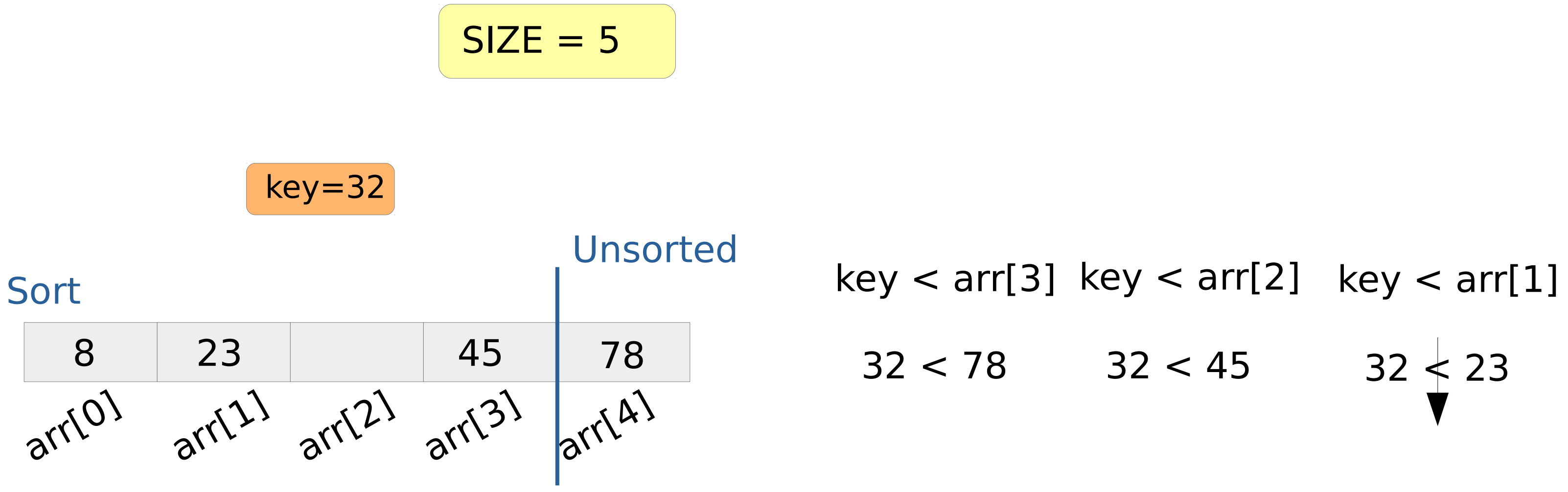
key < arr[3] key < arr[2]

32 < 78 32 < 45

Data Structure – Sorting Techniques

Insertion Sort

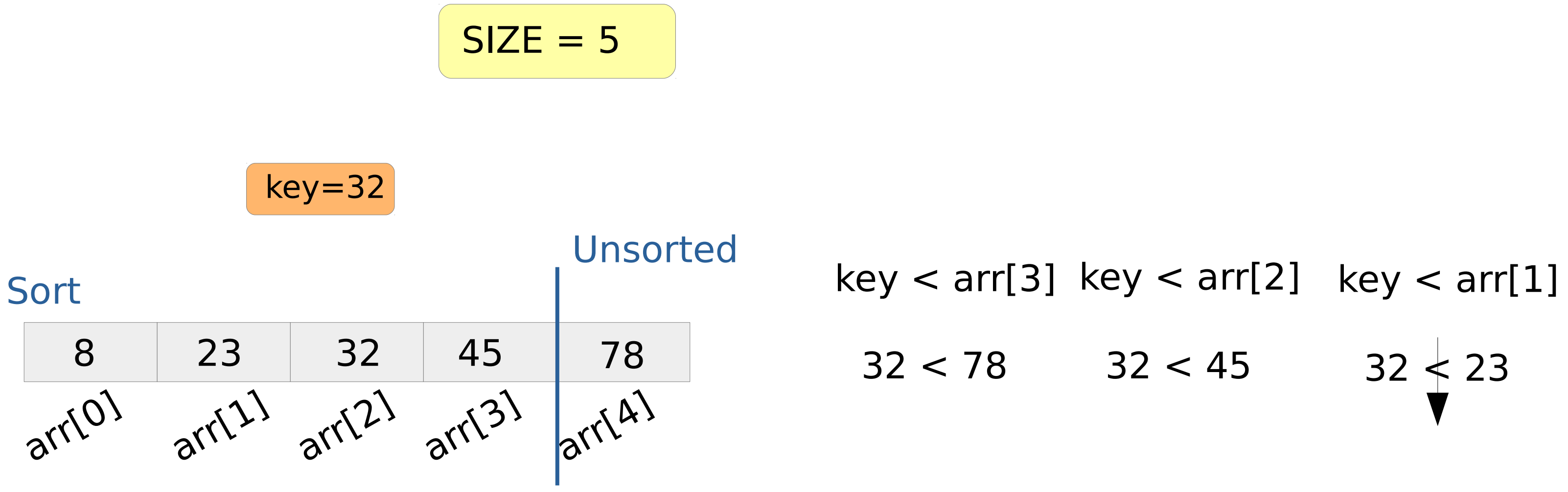
- arr[SIZE]



Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

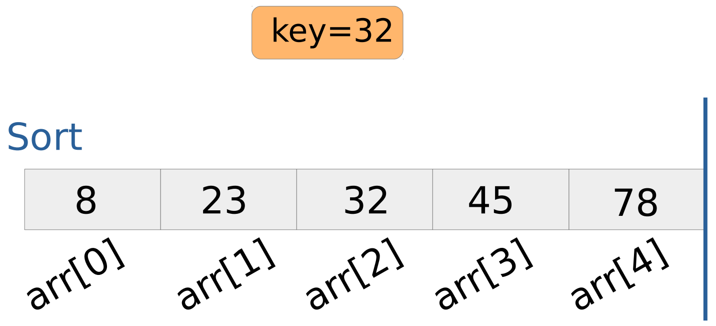


Data Structure – Sorting Techniques

Insertion Sort

- arr[SIZE]

SIZE = 5



key < arr[3]	key < arr[2]	key < arr[1]
32 < 78	32 < 45	32 < 23

Algorithm

Insertion Sort(arr,size)

```
for i = 1  upto  size
  key = arr[i]
  j = i - 1
  while ( j >=0 AND  arr[j] > key)
            arr[j+1] = arr[j]
            j = j - 1
  arr[j+1] = key
return e_true
```

Data Structure – Sorting Techniques

Insertion Sort

Advantages

- It exhibits a good performance when dealing with a small list.
- The insertion sort is an in-place sorting algorithm so the space requirement is minimal..

Insertion Sort

Advantages

- It exhibits a good performance when dealing with a small list.
- The insertion sort is an in-place sorting algorithm so the space requirement

  is minimal..

Disadvantages

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

CODE FORTHINGS

Code - Insertion Sort