Data Structure

Trainer : Nisha Dingare

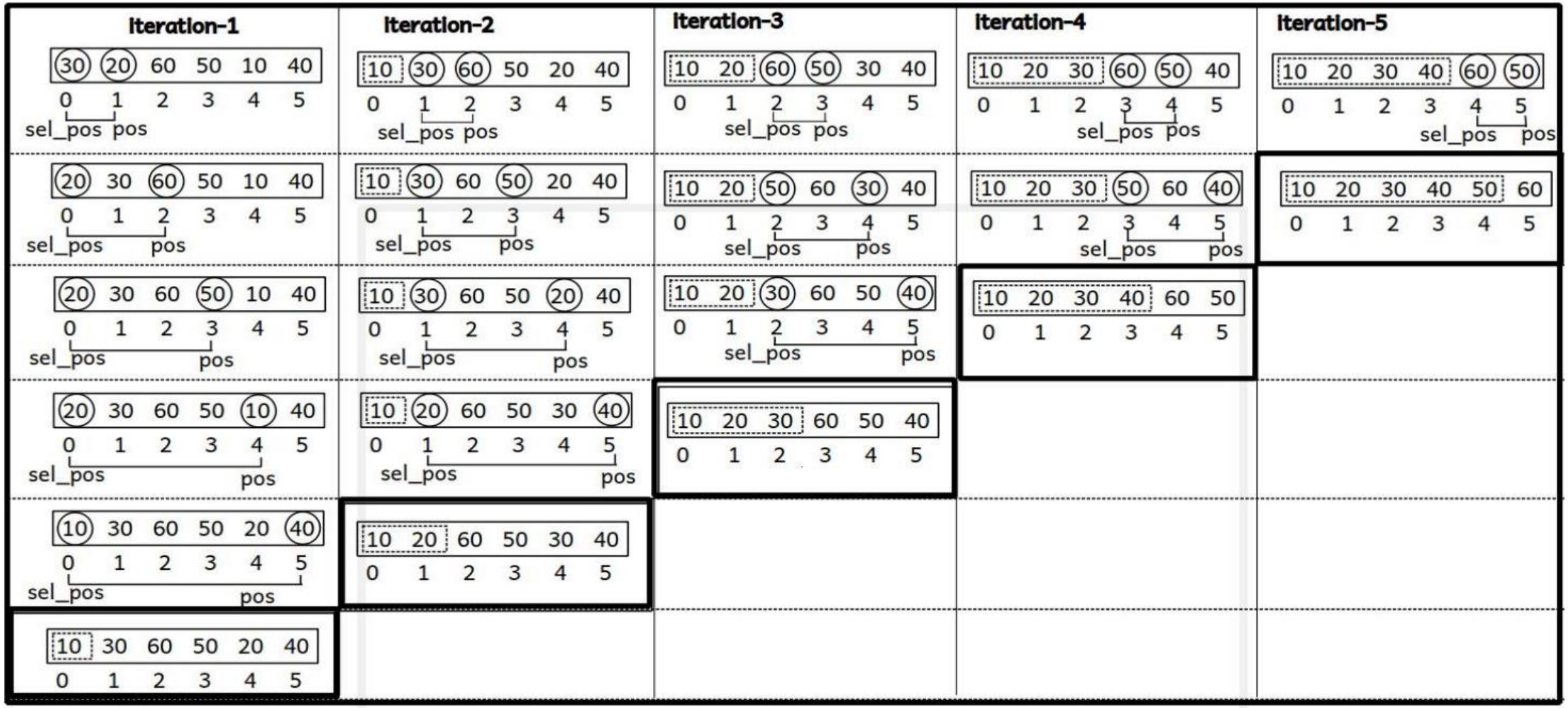
Email : nisha.dingare@sunbeaminfo.com

Sorting Algorithms :

- Sorting : Arranging data elements either in ascending or descending order. By default sorting takes place in ascending order.

1. **Selection Sort** :
   - In this algorithm, in first iteration, first position gets selected and element which is at selected position gets compared with all the elements after that. If selected position element found greater than any other position element then swapping takes place. At the end of first iteration smallest element gets settled at first position.
   - In the second iteration, second position gets selected and element which is at selected position gets compared with all elements after that. If selected position element found greater than any other position element then swapping takes place. At the end of second iteration second smallest element gets settled at second position, and so on in maximum (n-1) no. of iterations all array elements gets arranged in a sorted manner.
   - **Time complexity :**
     - Best Case : Ω(n2)
     - Worst Case : O(n2)
     - Average Case : θ(n2)

Selection Sort :



Bubble Sort :

- **Bubble Sort** :
  - In this algorithm, in every iteration, elements which are at two consecutive positions gets compared. If they are already in order then no need of swapping them, but if they are not in order i.e. if previous position element is greater than its next position element then swapping takes place.
  - By this logic in first iteration largest element gets settled at last position, in second iteration second largest element gets settled at second last position and so on, in max (n-1) no. of iterations all elements gets arranged in a sorted manner.

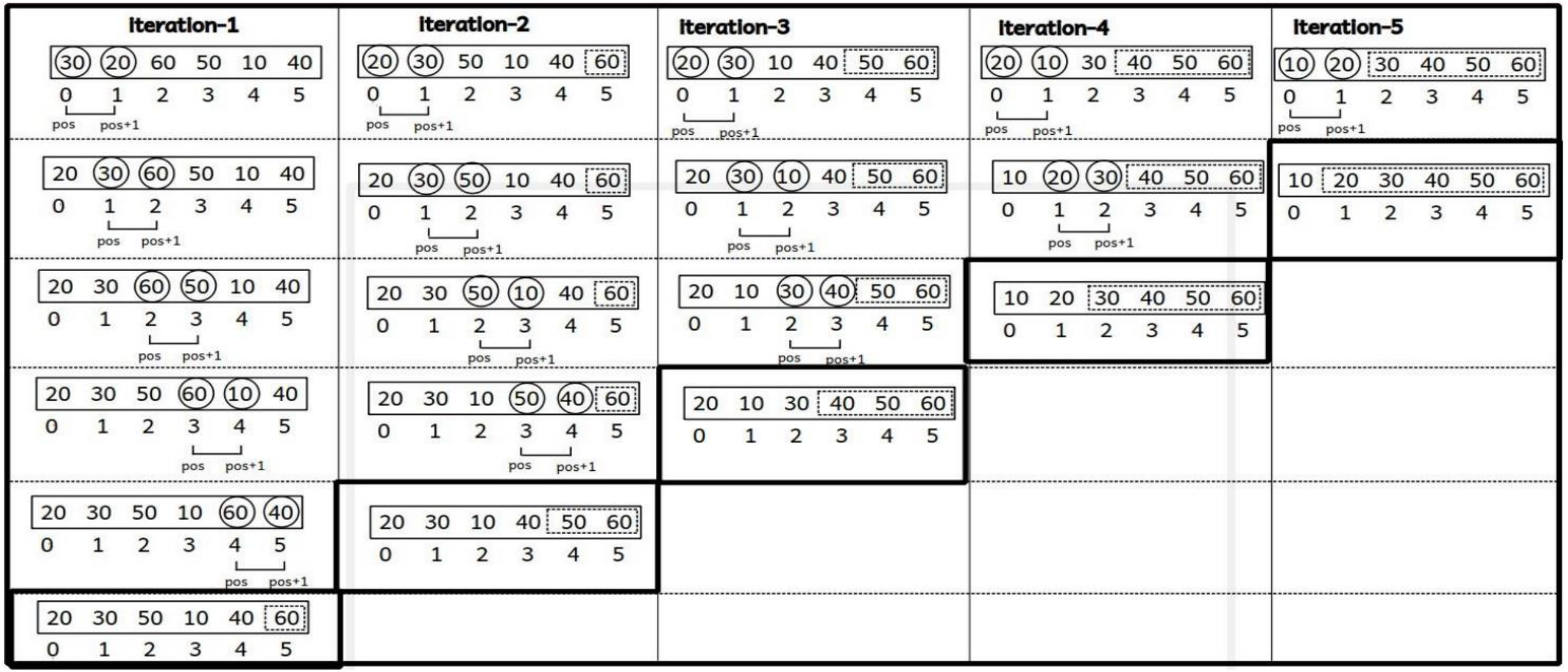
Time Complexity :

Best Case – if array elements are already arranged in a sorted manner : Ω(n)

Worst Case : O(n2)

Average Case : θ(n2)

Bubble Sort :


Iteration-1
30 20 60 50 10 40
0 1 2 3 4 5
pos pos+1
20 30 60 50 10 40
0 1 2 3 4 5
pos pos+1
20 30 60 50 10 40
0 1 2 3 4 5
pos pos+1
20 30 50 60 10 40
0 1 2 3 4 5
pos pos+1
20 30 50 10 60 40
0 1 2 3 4 5
pos pos+1
20 30 50 10 40 60
0 1 2 3 4 5
Iteration-2
20 30 50 10 40 60
0 1 2 3 4 5
pos pos+1
20 30 50 10 40 60
0 1 2 3 4 5
pos pos+1
20 30 50 10 40 60
0 1 2 3 4 5
pos pos+1
20 30 10 50 40 60
0 1 2 3 4 5
pos pos+1
20 30 10 40 50 60
0 1 2 3 4 5
Iteration-3
20 30 10 40 50 60
0 1 2 3 4 5
pos pos+1
20 30 10 40 50 60
0 1 2 3 4 5
pos pos+1
20 10 30 40 50 60
0 1 2 3 4 5
pos pos+1
20 10 30 40 50 60
0 1 2 3 4 5
Iteration-4
20 10 30 40 50 60
0 1 2 3 4 5
pos pos+1
10 20 30 40 50 60
0 1 2 3 4 5
pos pos+1
10 20 30 40 50 60
0 1 2 3 4 5
Iteration-5
10 20 30 40 50 60
0 1 2 3 4 5
pos pos+1
10 20 30 40 50 60
0 1 2 3 4 5

Insertion Sort :

- In this algorithm, in every iteration one element gets selected as a key element and key element gets inserted into an array at its appropriate position in a such a way that elements which are at left side are arranged in a sorted manner, and so on. In max (n-1) no. of iterations all array elements gets arranged in a sorted manner.
- This algorithm works efficiently for already sorted input sequence by design and hence running time of an algorithm is O(n) and it is considered as a best case.
- Time Complexity :
  - Best Case Worst Case : Ω(n) – if array elements are already arranged in a sorted manner.
  - Worst Case : O(n2)
  - Average Case: θ(n2)
- Insertion sort algorithm is an efficient algorithm for smaller input size array

Thank You !!