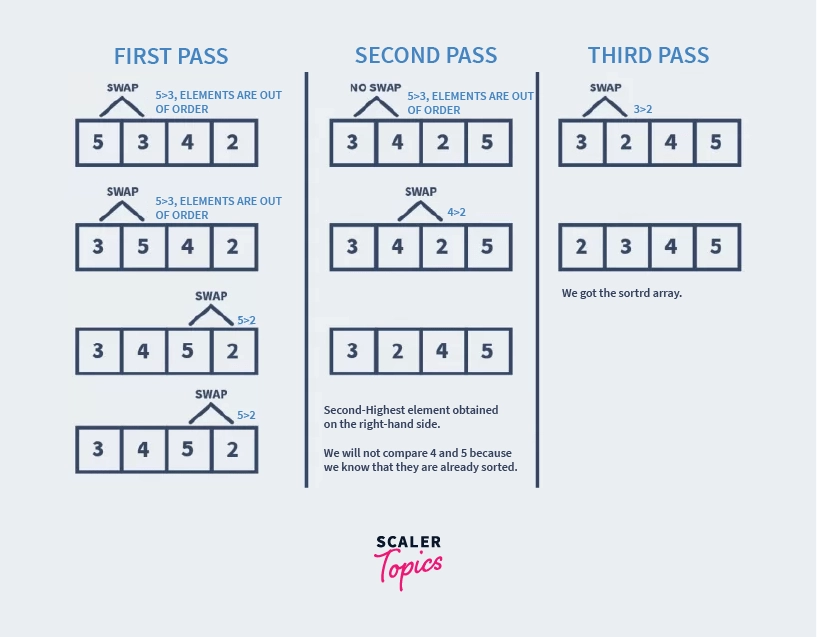
**Lab#04**

Objective: Implement sorting algorithms (Array)

**Bubble Sort**

****

**Pseudo Code**

|  |
| --- |
| for i=0 to A.length-2  for j=0 to A.length-2-i  if(A[j]>A[j+1])  temp=A[j+1]  A[j+1]=A[j]  A[j]=temp |

**Selection Sort**



**Pseudo Code**

|  |
| --- |
| for i=0 to A.length-2  mindIndex=i  for j=i+1 to A.length-1  if(A[j]<A[minIndex])  minIndex=j  temp=A[i]  A[i]=A[minIndex]  A[minIndex]=temp |

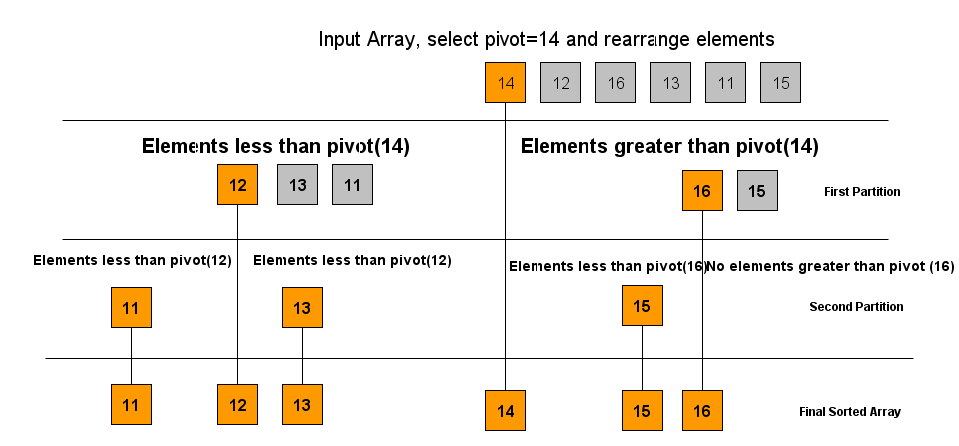
**Insertion Sort**



**Pseudo Code**

|  |
| --- |
| INSERTION-SORT(A)  for i = 1 to n  current ← A [i]  j ← i – 1  while j > = 0 and A[j] > current  A[j+1] ← A[j]  j ← j – 1  End while  A[j+1] ← current  End for |

**Quick Sort**



**Partition Alog**

**Diagram

Description automatically generated**

**Quick Sort Algo**

**Text, letter

Description automatically generated**

**A picture containing letter

Description automatically generated**

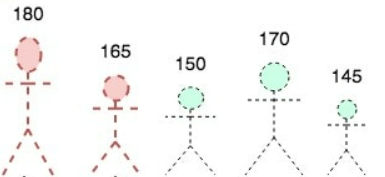
**How to calculate the execution time(milliseconds) of a method in java:**

|  |
| --- |
| long before1=System.currentTimeMillis();    Searching\_lab.greeting1();    long after1 = System.currentTimeMillis();  System.out.println("Execution time of greeting1 method is : "+(after1-before1)); |

|  |
| --- |
| long before1=System.nanoime();    Searching\_lab.greeting1();    long after1 = System. System.nanoime();  System.out.println("Execution time of greeting1 method is : "+(after1-before1)); |

**Exercises**

**Task#01**

****

Sort the given 05 students in ascending order of their heights

1. **Bubble sort**

//Create method BubbleSort1D(int[] A)

1. **Selection sort**

//Create method SelectionSort1D(int[] A)

1. **Insertion sort**

//Create method InsertionSort1D(int[] A)

1. **Quick sort**

//Create method QuickSort1D(int[] A)

1. **Display the execution time of sorting algos and examine which one is the fastest and explain why?**

Sample Output:

Time took by bubble sort : \_\_\_\_ (time in milliseconds)

Time took by selection sort : \_\_\_\_ (time in milliseconds)

Time took by insertion sort : \_\_\_\_ (time in milliseconds)

Time took by quick sort : \_\_\_\_ (time in milliseconds)

**Task#02**

Sort the following 2D array using (bubble, selection, insertion, and quick sort)

|  |  |  |
| --- | --- | --- |
| **2** | **10** | **15** |
| **5** | **1** | **3** |
| **6** | **9** | **4** |

**Submit your lab#04 work by today on Teams**