Algorithm:

1. From the list of elements choose any element as the pivot (here the middle element).
2. After selecting the pivot, divide the array into two parts. Here the division is known as partitioning.
3. Partitioning doesn't mean that dividing the array or the list of elements into two halves. The array is divided in such a way that the elements greater than the pivot will be to the right of the pivot and the elements lesser than the pivot will be to the left of it.
4. The pivot element is in its sorted position whereas the elements to the right and left to it are unsorted.
5. Again, we implement the partition on the left and right sub-arrays of the pivot recursively.

Pseudocode:

```
start procedure
partition(arr,l,h)
  pivot ← arr[(l+h)/2]
  i←l
  j←h
  while i<j do
    if arr[i]<=pivot then
      i←i+1
    end if
    if arr[j]>pivot then
      j←j-1
    end if
    if i<j then
      temp←arr[l]
      arr[i]←arr[j]
      arr[j]←temp
    end if
  end while
  temp←arr[l]
  arr[l]←arr[j]
  arr[j]←temp
end procedure
start procedure
quicksort(arr,l,h)
  if l<h then
    r←partition(arr,l,h)
    quicksort(arr,l,r)
    quicksort(arr,r+1,h)
  end if
end procedure
```

Implementation of Quick sort in Java:

```
import java.util.*;
class QuickSort {

   static int n,i,j,temp,step=0;
   static Scanner sc =new Scanner(System.in);
   static int partition(int [] arr,int l,int h){
      step++;
      int pivot=arr[(l+h)/2];
      i=l;j=h;
      while(i<j){
         if(arr[i]<=pivot)
            i++;
         if(arr[j]>pivot)
            j--;
         if(i<j){
            temp=arr[i];
            arr[i]=arr[j];
            arr[j]=temp;
         }
      }
      temp=arr[l];
      arr[l]=arr[j];
      arr[j]=temp;
      System.out.print("\n At " + step + " the arrangement is: ");
      for(i=0;i<n;i++)
         System.out.print(arr[i]);
      return j;
   }

   static void quicksort(int [] arr,int l,int h){
      if(l<h){
         int r=partition(arr,l,h);
         quicksort(arr,l,r);
         quicksort(arr,r+1,h);
      }
   }
   public static void main(String [] args){
      System.out.println(“File downloaded from Techprodezza”);
      System.out.println("Enter the size of the array: ");
      n=sc.nextInt();
      int[] arr=new int[n];
      System.out.println("Enter the elements of the array: ");
      for(i=0;i<n;i++)
         arr[i]=sc.nextInt();
      quicksort(arr, 0, n-1);
```

```
        System.out.println("\n The ordered array of elements is: ");
        for(i=0;i<n;i++)
            System.out.print(" " +arr[i]);
    }
}
```

Notations:

arr → Array of elements to be sorted
i,j → Variables used in loops
l → least index of array
h → highest index of array
pivot → element selected to sort
r → the index at which the pivot is placed or
  the sorted index of pivot
temp → temporary variable to swap elements
partition function → function used to section the elements
quicksort function → function used to sort the elements;
      recursive use.