Quick Sort In Java

```
public class Main{

   public static void main(String args[])
   {
          // quick sort = moves smaller elements to left of a pivot.
          //                                  recursively divide array in 2 partitions

          //                 run-time complexity = Best case O(n log(n))
          //                                                            Average case O(n log(n))
          //                                                            Worst case O(n^2) if already sorted

          //                 space complexity    = O(log(n)) due to recursion

      int[] array = {8, 2, 4, 7, 1, 3, 9, 6, 5};

      quickSort(array, 0, array.length - 1);

      for(int i : array){
         System.out.print(i + " ");
      }
   }

          private static void quickSort(int[] array, int start, int end) {

                    if(end <= start) return; //base case

                    int pivot = partition(array, start, end);
                    quickSort(array, start, pivot - 1);
                    quickSort(array, pivot + 1, end);
          }
          private static int partition(int[] array, int start, int end) {

                    int pivot = array[end];
                    int i = start - 1;

                    for(int j = start; j <= end; j++) {
                              if(array[j] < pivot) {
                                        i++;
                                        int temp = array[i];
                                        array[i] = array[j];
                                        array[j] = temp;
                              }
                    }
                    i++;
                    int temp = array[i];
                    array[i] = array[end];
                    array[end] = temp;
```

```
        return i;
    }
}
```