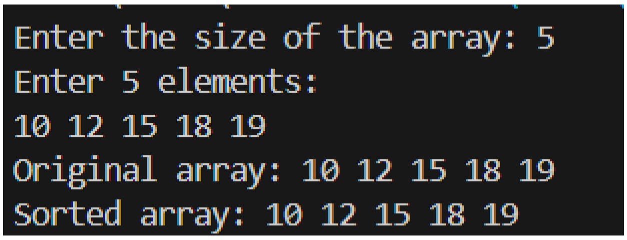
QuickSort

```
import java.util.Scanner;
public class QuickSort {
  public static int partition(int[] arr, int low, int high) {
        int pivot = arr[high];
        int i = low - 1;
 for (int j = low; j < high; j++) {
            if (arr[j] <= pivot) {
                i++;
                     int temp = arr[i];
                arr[i] = arr[j];
                arr[j] = temp;
            }
        }
 int temp = arr[i + 1];
        arr[i + 1] = arr[high];
        arr[high] = temp;
 return i + 1;
    }
public static void quickSort(int[] arr, int low, int high) {
        if (low < high) {
            int pivotIndex = partition(arr, low, high);
            quickSort(arr, low, pivotIndex - 1);
            quickSort(arr, pivotIndex + 1, high);
        }
    }
  public static void main(String[] args) {
        Scanner scanner = new Scanner(System.in);
 System.out.print("Enter the size of the array: ");
        int size = scanner.nextInt();
  int[] arr = new int[size];
System.out.println("Enter " + size + " elements:");
        for (int i = 0; i < size; i++) {
            arr[i] = scanner.nextInt();
    }
  System.out.print("Original array: ");
                     for (int num : arr) {
            System.out.print(num + " ");
        }
        System.out.println();
    quickSort(arr, 0, size - 1);
System.out.print("Sorted array: ");
        for (int num : arr) {
            System.out.print(num + " ");
        } }}
```

Output:



```
Enter the size of the array: 5
Enter 5 elements:
10 12 15 18 19
Original array: 10 12 15 18 19
Sorted array: 10 12 15 18 19
```

Analysis of QuickShort

In this code implements the QuickSort algorithm, which sorts an array by recursively partitioning it around a pivot.

- partition(): Rearranges elements based on the pivot.
- quickSort(): Recursively sorts the array.
- main(): Takes user input, sorts the array, and displays the result.