Merge Sort:

package arrays.sorting;

import java.util.\*;

public class mergeSort {

    static void merge(int a[], int x, int y, int r) {

        int n1 = y - x + 1;

        int n2 = r - y;

        int a1[] = new int[n1];

        int a2[] = new int[n2];

        for (int i = 0; i < n1; i++)

            a1[i] = a[x + i];

        for (int j = 0; j < n2; j++)

            a2[j] = a[y + 1 + j];

        int i, j, k;

        i = 0;

        j = 0;

        k = x;

        while (i < n1 && j < n2) {

            if (a1[i] <= a2[j]) {

                a[k] = a1[i];

                i++;

            } else {

                a[k] = a2[j];

                j++;

            }

            k++;

        }

        while (i < n1) {

            a[k] = a1[i];

            i++;

            k++;

        }

        while (j < n2) {

            a[k] = a2[j];

            j++;

            k++;

        }

    }

    static void sort(int a[], int l, int r) {

        if (l < r) {

            int m = (l + r) / 2;

            sort(a, l, m);

            sort(a, m + 1, r);

            merge(a, l, m, r);

        }

    }

    static void printArray(int arr[]) {

        int n = arr.length;

        for (int i = 0; i < n; ++i)

            System.out.print(arr[i] + " ");

        System.out.println();

    }

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter limit: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter array[" + n + "]: ");

        for (int i = 0; i < arr.length; i++) {

            arr[i] = sc.nextInt();

        }

        sort(arr, 0, arr.length - 1);

        System.out.println("Sorted array:");

        printArray(arr);

    }

}

Graphical user interface, application

Description automatically generated with medium confidence

Quick sort:

package arrays.sorting;

import java.util.\*;

class quick {

    static int partition(int array[], int low, int high) {

        int pivot = array[high];

        int i = (low - 1);

        for (int j = low; j < high; j++) {

            if (array[j] <= pivot) {

                i++;

                int temp = array[i];

                array[i] = array[j];

                array[j] = temp;

            }

        }

        int temp = array[i + 1];

        array[i + 1] = array[high];

        array[high] = temp;

        return (i + 1);

    }

    static void quickSort(int array[], int low, int high) {

        if (low < high) {

            int pi = partition(array, low, high);

            quickSort(array, low, pi - 1);

            quickSort(array, pi + 1, high);

        }

    }

    static void print(int[] a) {

        for (int i = 0; i < a.length; i++) {

            System.out.print(a[i] + " ");

        }

    }

    public static void main(String args[]) {

        Scanner sc = new Scanner(System.in);

        System.out.println("Enter limit: ");

        int n = sc.nextInt();

        int arr[] = new int[n];

        System.out.println("Enter array[" + n + "]: ");

        for (int i = 0; i < arr.length; i++) {

            arr[i] = sc.nextInt();

        }

        System.out.println("Initial Array");

        print(arr);

        int size = arr.length;

        quickSort(arr, 0, size - 1);

        System.out.println("\n Sorted Array: ");

        print(arr);

    }

}

Graphical user interface, text, application

Description automatically generated