**Quick Sort:**

package practice;

import java.util.Scanner;

public class QuickSort{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int[] arr = new int[n];

for(int i=0;i<n;i++) {

arr[i] = sc.nextInt();

}

quickSort(arr,0,n-1);

for(int i=0;i<n;i++) {

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

}

}

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

if(low<high) {

int p = partition(arr,low,high);

quickSort(arr,low,p-1);

quickSort(arr,p+1,high);

}

}

private static int partition(int arr[],int low,int high) {

int start = low+1, end = high, pivot = low;

while(start<end) {

while(arr[start]<=arr[pivot]&&start<=high) {

start++;

}

while(arr[end]>arr[pivot]&&end>=low) {

end--;

}

if(start<end) {

int temp = arr[start];

arr[start] = arr[end];

arr[end] = temp;

}

}

if(start>end) {

int temp = arr[pivot];

arr[pivot] = arr[end];

arr[end] = temp;

}

return end;

}

}

**Merge Sort:**

package practice;

import java.util.Scanner;

public class MergeSort{

public static void main(String[] args) {

Scanner sc = new Scanner(System.in);

int n = sc.nextInt();

int[] arr = new int[n];

for(int i=0;i<n;i++) {

arr[i] = sc.nextInt();

}

arr = mergeSort(arr,0,n-1);

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

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

}

}

private static int[] mergeSort(int[] arr, int i, int n) {

if(i>=n) {

return new int[] {arr[i]};

}

int mid = (i+n)/2;

int[] arr1 = mergeSort(arr,i,mid);

int[] arr2 = mergeSort(arr,mid+1,n);

arr = merge(arr1,arr2);

return arr;

}

private static int[] merge(int[] arr1, int[] arr2) {

int[] ans = new int[arr1.length+arr2.length];

int i=0,j=0,k=0;

while(i<arr1.length&&j<arr2.length) {

if(arr1[i]<arr2[j]) {

ans[k] = arr1[i];

i++;

k++;

}

else {

ans[k] = arr2[j];

j++;

k++;

}

}

while(i<arr1.length) {

ans[k]= arr1[i];

i++;

k++;

}

while(j<arr2.length) {

ans[k]= arr2[j];

j++;

k++;

}

return ans;

}

}