1. **Selection Sort**

import java.util.Scanner;

public class selectionSort {

static int compareCount=0,swapCount=0;

public static int[] selectionSort(int a[]){

int n=a.length,t,temp;

for(int j=n-1;j>=0;j--){

t=0;

for(int k=1;k<=j;k++){

compareCount++;

if(a[t]<a[k])

t=k;

}

temp=a[j];

a[j]=a[t];

a[t]=temp;

swapCount++;

}

return a;

}

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int a[]=new int[1000];

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

a[i]=(int)(Math.random()\*1000+1);

long startTime = System.currentTimeMillis();

a=selectionSort(a);

long stopTime = System.currentTimeMillis();

long elapsedTime = stopTime - startTime;

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

System.out.println((i+1)+": "+a[i]+" ");

System.out.println("\nTime elapsed for selection sorting "+a.length+" integers: "+elapsedTime+" miliseconds");

System.out.println("Comparison count: "+compareCount);

System.out.println("Swap count: "+swapCount);

}

}

1. **Merge Sort**

import java.util.Scanner;

public class MergeSort {

static int compareCount=0,swapCount=0;

public static int[] mergeSort(int a[],int p, int r){

if(p<r){

int q=(p+r)/2;

mergeSort(a,p,q);

mergeSort(a,q+1,r);

merge(a,p,q,r);

}

return a;

}

public static void merge(int a[],int p,int q,int r){

int i,j,k;

int n1=q-p+1;

int n2=r-q;

int left[]=new int[n1];

int right[]=new int[n2];

for(i=0;i<n1;i++)

left[i]=a[p+i];

for(j=0;j<n2;j++)

right[j]=a[q+j+1];

i=0;

j=0;

k=p;

while(i<n1 && j<n2){

if(left[i]<=right[j]){

a[k]=left[i];

i++;

}

else

{

a[k]=right[j];

j++;

}

k++;

compareCount++;

}

while(i<n1){

a[k]=left[i];

k++;

i++;

}

while(j<n2){

a[k]=right[j];

k++;

j++;

}

}

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int a[]=new int[1000];

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

a[i]=(int)(Math.random()\*1000+1);

long startTime = System.currentTimeMillis();

a=mergeSort(a,0,a.length-1);

long stopTime = System.currentTimeMillis();

long elapsedTime = stopTime - startTime;

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

System.out.println((i+1)+": "+a[i]+" ");

System.out.println("\nTime elapsed for merge sorting "+a.length+" integers: "+elapsedTime+" miliseconds");

System.out.println("Comparison count: "+compareCount);

}

}

1. **Quick Sort**

import java.util.Scanner;

public class QuickSort {

static int compareCount=0,swapCount=0;

public static int[] quickSort(int a[],int first,int last){

int i,j,temp;

if(first<last){

i=first+1;

while(a[i]<a[first]){

i++;

}

j=last;

while(a[j]>a[first]){

j--;

}

while(i<j)

{

temp=a[i];

a[i]=a[j];

a[j]=temp;

swapCount++;

while(a[i]<=a[first]){

i++;

compareCount++;

}

while(a[j]>a[first]){

j--;

compareCount++;

}

}

temp=a[first];

a[first]=a[j];

a[j]=a[first];

swapCount++;

quickSort(a,first,j-1);

quickSort(a,j+1,last);

}

return a;

}

public static void main(String args[]){

Scanner sc=new Scanner(System.in);

int a[]=new int[1000];

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

a[i]=(int)(Math.random()\*1000+1);

long startTime = System.currentTimeMillis();

a=quickSort(a,0,a.length-1);

long stopTime = System.currentTimeMillis();

long elapsedTime = stopTime - startTime;

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

System.out.println((i+1)+": "+a[i]+" ");

System.out.println("\nTime elapsed for quick sorting "+a.length+" integers: "+elapsedTime+" miliseconds");

System.out.println("Comparison count: "+compareCount);

System.out.println("Swap count: "+swapCount);

}

}