// Vtricks Edtech

//Vinay Haritsa

// Documented for VTU Students

//import libraries

**import** java.util.Random;

**public** **class** MergeSort

{

**public** **static** **void** divide(**int**[] a, **int** n)

{

**if** (n < 2) {

**return**;

}

**int** mid = n / 2; // find out middle element

**int**[] l = **new** **int**[mid]; // create array of left array

**int**[] r = **new** **int**[n - mid]; // create array of right array

// why n-mid because if n=7 , middle =3 and n- mid = 4 so

// l array has 3 elements and r array has 4 elements

// this loop will create left array by pulling all elements before mid element

//and will put it to l array

**for** (**int** i = 0; i < mid; i++) {

l[i] = a[i];

}

// this loop will create right array by pulling all elements after mid element

//and will put it to r array

**for** (**int** i = mid; i < n; i++) {

r[i - mid] = a[i];

}

//call divide function for left and right

*divide*(l, mid);

*divide*(r, n - mid);

// call merge function and pass the original array , left array

//right array , number of elements before middle position and

//number of elements after middle position

*merge*(a, l, r, mid, n - mid);

}

//this function is used to merge 2 arrays

**public** **static** **void** merge(**int**[] a, **int**[] l, **int**[] r, **int** left, **int** right)

{

// this function has 3 loops

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

//loop 1 to check both left and right simultaneously

**while** (i < left && j < right) {

**if** (l[i] <= r[j]) {

a[k++] = l[i++];

}

**else** {

a[k++] = r[j++];

}

}

//loop2 to check for left array

**while** (i < left) {

a[k++] = l[i++];

}

//loop3 to check for right array

**while** (j < right) {

a[k++] = r[j++];

}

System.***out***.println("element in sorted order is ");

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

{

System.***out***.println(a[num]);

}

}

//main function

**public** **static** **void** main(String[] args) {

/\*\* int[] actual = { 5, 1, 6, 2, 3, 4 };

divide(actual, actual.length);

//divide(actual, actual.length);

\*\*/

**int**[] numbers = **new** **int**[5500];

Random generator = **new** Random();

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

numbers[i] = generator.nextInt(1000);

}

**long** startTime = System.*currentTimeMillis*();

*divide*(numbers, numbers.length);

**long** stopTime = System.*currentTimeMillis*();

**long** elapsedTime = stopTime - startTime;

System.***out***.println("Mergesort time in miliseconds " + elapsedTime);

}

}