**import** java.util.Arrays;

**import** java.util.Random;

**public** **class** InsertionRunning {

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

//initialize array. length should be changed manually when producing the diagram.

**int**[] array= **new** **int**[1000];

//create a random array

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

//use java.random to generate each element in the array.

Random random = **new** Random();

//because we want the range to start from 0 to the length to the array;

array[i]= random.nextInt(array.length);

}

//clone the random array for method2.

**int**[] arrayClone = array.clone();

//print it out

System.*out*.println("The original array is:");

System.*out*.println(Arrays.*toString*(array));

//method1: the java built-in sorting method.

**long** start1 = System.*currentTimeMillis*();

Arrays.*sort*(array); //built-in method

**long** end1 = System.*currentTimeMillis*();

// System.out.println(Arrays.toString(array));

**long** duration1 = end1 - start1;

System.*out*.println("Sorting by java built-in method took " + duration1 + " ms");

//method2: which used clone of the original array.

**long** start2 = System.*currentTimeMillis*();

*insertionSort*(arrayClone); // Task we want to benchmark

**long** end2 = System.*currentTimeMillis*();

// System.out.println(Arrays.toString(array));

**long** duration2 = end2 - start2;

System.*out*.println("Sorting by insertionSort took " + duration2 + " ms");

}

**public** **static** **void** insertionSort (**int** [] a) {

**for** (**int** i = 1; i < a.length; i++) {

// Invariant: a[0..i-1] is already sorted.

**final** **int** cur = a[i];

// Now insert cur into a[0..i-1],

// shifting greater elements upward.

**int** j = i;

**while** (j > 0 && a[j-1] > cur)

a[j] = a[--j];

a[j] = cur;

}

}

}