Topic 1 Discussion 1

Consider an array of 100 integers, which are completely unsorted. Provide a code snippet that will find the largest number in the array and calculate the number of computational steps required. Create a Loom video in which you comment on your code and give your assessment of the computational time required. Paste the link to your video here.

public class wk1dq1 {

//driver

public static void main(String[] args) {

int[] rcArray = makeRandomArray();

findLargestNumber(rcArray);

}

//call to make array

public static int[] makeRandomArray() {

Random rm = new Random(); // create random object

int[] rcArray = new int[100]; // sets 100 random integers

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

rcArray[i] = rm.nextInt(1000); // storing random integers in an array max value of 1000

System.out.println(rcArray[i]);// prints 100 integer random array

}

return rcArray;

}

//call to find largest number

public static void findLargestNumber(int[] rcArray) {

int largestNumber = 0;

int count = 0;

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

if (rcArray[i] > largestNumber) {

largestNumber = rcArray[i];// this takes the largest number in the array and compares it to the other 99

// to ensure it is the largest. largest number at start is set at 0

count++;// adds 1 to count

} else if (rcArray[i] < largestNumber) {

count++;// adds 1 to count

}

}

System.out.println("The largest number is " + largestNumber + ". Taking " + count + " times to compute. ");

}

}

Here is my attempt at finding the largest number in a random array of 100 integers. I tried to make it as simple as possible, with the lease amount of steps.

Here is the Loom video explaining the code and running the program.

https://www.loom.com/share/190286e4098949e089d1d10d5268a429