/\* Amir Yamini

\* Professor Schwartz

\* 09-26-2017

\* Project 1 - SortsTester

\*/

import java.util.Random;

import java.util.Timer;

public class SortsTester {

public static void main (String args[])

{

Sorts soides = new Sorts();

/\*

int array1[] = { 34, 67, 23, 19, 122, 300, 2, 5, 17, 18, 5, 4, 3, 19, -40, 23 };

int array2[] = { 34, 67, 23, 19, 122, 300, 2, 5, 17, 18, 5, 4, 3, 19, -40, 23 };

//MERGESORT

soides.mergeSort(array1, 0, array1.length-1);

System.out.print("Merge Sorted array: [");

for (int x = 0; x < array1.length; x++)

{

System.out.print(array1[x]);

if(x+1 != array1.length) {

System.out.print(", ");

}

}

System.out.println("]");

System.out.println("mergeSort succeeded in sorting this array: " + soides.isSorted(array1));

//////////////////////////////////////////////////////////////////

//QUICKSORT

soides.quickSort(array2, 0, array2.length-1);

System.out.print("Quick Sorted array: [");

for (int y = 0; y < array2.length; y++)

{

System.out.print(array2[y]);

if(y+1 != array2.length) {

System.out.print(", ");

}

}

System.out.println("]");

System.out.println("quickSort succeeded in sorting this array: " + soides.isSorted(array2));

\*/

/////////////////////////////////////////////////////////////////

//BOTH

long start;

long end;

long timeM;

long timeQ;

int trialCount = 0;

int quickCount = 0;

int mergeCount = 0;

long quickRun = 0;

long mergeRun = 0;

long qAverage = 0;

long mAverage = 0;

int arrayCount = 0;

int arraySize[] = {10,100,1000,10000,100000,1000000,2000000};

while(arrayCount < 7) {

int rand1[] = new int[arraySize[arrayCount]];

int rand2[] = new int[arraySize[arrayCount]];

System.out.println(arrayCount);

trialCount = 0;

mergeRun = 0;

quickRun = 0;

mergeCount = 0;

quickCount = 0;

while(trialCount < 20) {

//MERGESORT

for(int i = 0; i < arraySize[arrayCount]; i++) {

rand1[i] = (int) (Math.random() \* 1000000 + 1);

rand2[i] = rand1[i];

}

start = System.nanoTime();

soides.mergeSort(rand1, 0, arraySize[arrayCount]-1);

end = System.nanoTime();

timeM = end - start;

//System.out.println(timeM);

//QUICKSORT

start = System.nanoTime();

soides.quickSort(rand2, 0, arraySize[arrayCount]-1);

end = System.nanoTime();

timeQ = end - start;

//System.out.println(timeQ);

mergeRun += timeM;

quickRun += timeQ;

if(timeM > timeQ) {

//System.out.println("QuickSort Won!");

quickCount++;

//System.out.println(quickCount);

}else {

//System.out.println("MergeSort Won!");

mergeCount++;

//System.out.println(mergeCount);

}

trialCount++;

}// trials loop

mAverage = mergeRun/20;

qAverage = quickRun/20;

System.out.println(arraySize[arrayCount]);

System.out.println("Merge wins: " + mergeCount + "," + "Quick wins: " + quickCount);

System.out.println("Merge Average: " + mAverage + ", " + "Quick Average: " + qAverage);

System.out.println("The mean MergeSort runtime is: " + mAverage / (arraySize[arrayCount] \* (Math.log(arraySize[arrayCount]) / Math.log(2))));

System.out.println("The mean QuickSort runtime is: " + qAverage / (arraySize[arrayCount] \* (Math.log(arraySize[arrayCount]) / Math.log(2))));

arrayCount++;

}

}

}