/\* Amir Yamini

\* Professor Schwartz

\* 09-26-2017

\* Project 1 - Sorts

\*/

public class Sorts {

public static void merge(int a[], int i, int j, int k) {

int half1 = j - i + 1; //for left sub-array

int half2 = k - j; //for right sub-array

//split the array sizes in half

int array1[] = new int[half1];

int array2[] = new int[half2];

//store the half arrays into arrays

for (int p = 0; p < half1; ++p)

array1[p] = a[i + p];

for (int t = 0; t < half2; ++t)

array2[t] = a[j + 1 + t];

int x = 0, y = 0;

int z = i;

while (x < half1 && y < half2) { //while pointer is less than length of the half array

if (array1[x] <= array2[y]) { //find smaller of the two elements and put it in the main array

a[z] = array1[x];

x++;

} else {

a[z] = array2[y];

y++;

}

z++; //go to next cell in main array

}

//these next two while loops will ensure that no left over elements in either half array are missed

while (x < half1) {

a[z] = array1[x];

x++;

z++;

}

while (y < half2) {

a[z] = array2[y];

y++;

z++;

}

}

public static void mergeSort(int[] a ) {

int length = a.length-1;

mergeSort(a,0,length);

}

public static void mergeSort(int []a, int i, int k) { // sort a[i...k]

int m;

if(i<k) {

m = (i+k)/2;

mergeSort(a,i,m);

mergeSort(a,m+1,k);

merge(a,i,m,k);

}

}

public static int partition (int []s, int a, int b) {

int p = s[b]; //this is your pivot

int l = a;

int r = b-1;

int temp; //temp will assist in swap

while (l <= r) {

while (l <= r && s[l] <= p) {

l++;

}

while (r >= l && s[r] >= p) {

r--;

}

if(l < r) {

//swap

temp = s[l];

s[l] = s[r];

s[r] = temp;

}

}

//swap

temp = s[l];

s[l] = s[b];

s[b] = temp;

return l;

}

public static void quickSort (int []a, int i, int j) {

if (i<j) {

int s = partition(a, i, j);

quickSort(a, i, s-1);

quickSort(a, s+1, j);

}

}

public static void quickSort(int[] a ) {

int length = a.length-1;

quickSort(a,0,length);

}

//check to see if the functions are actually sorted in incrementing order

public static boolean isSorted(int [] a){

for ( int i = 0; i< a.length-1; i++){

if(a[i]>a[i+1]){

return false;

}

}

return true;

}

}