package PRELIM;

import java.util.\*;

public class BucketSort2 {

public static void main(String[] args) {

ArrayList arr = new ArrayList();

arr.addAll(Arrays.asList(1,100,8,23,34,456));

System.out.println("final result " + bucketSort(arr));

}

public static ArrayList bucketSort(ArrayList arr){

return bucketSort(arr, 0);

}

private static ArrayList bucketSort(ArrayList arr, int index){

ArrayList<ArrayList<String>> buckets = new ArrayList<>();

ArrayList result = new ArrayList();

// ----------FORMAT NUMBERS TO BE 3 DIGIT-----------

Object obj;

for (int i = 0; i < arr.size(); i++){

obj = arr.get(i);

arr.remove(i);

try {

arr.add(i, String.format("%03d", obj));

}

catch (Exception e){

arr.add(obj);

}

}

String str, str2;

int num, peek, bucketindex;

for (int k = 0; k < arr.size(); k++){

str = (String)arr.get(k);

num = Integer.parseInt(str.charAt(index) + "");

// -------------INITIAL CASE----------------

if (buckets.size() == 0) {

ArrayList al = new ArrayList();

al.add(str);

buckets.add(al);

}

else {

bucketindex = 1;

for (int j = 0; j < buckets.size(); j++){

str2 = buckets.get(j).get(0);

peek = Integer.parseInt(str2.charAt(index) + "");

// -------------BUCKET INDEX - > position of the bucket in the buckets-----

if (num < peek) {

bucketindex = 0;

break;

}

else if (num > peek) {

bucketindex++;

if (bucketindex == buckets.size())

break;

}

}

// -----------if bucketindex is more than the current size

// ex, a new number is introduced, create a new bucket--------

if (bucketindex >= buckets.size()) {

buckets.add(new ArrayList<>(Collections.singletonList(str)));

}

// -------------get the bucket from the buckets and add the number--------

else{

buckets.get(bucketindex).add(str);

}

}

}

System.out.println("buckets: "+buckets);

// --------CHECK THE BUCKETS----------------

for (int i = 0; i < buckets.size(); i++){

ArrayList bucket = buckets.get(i);

// -------BASE CASE SCENARIO

// merge the buckets which are only containing 1 item---------

if (bucket.size() == 1){

result = merge(result, bucket);

}

// --------if the bucket has more than 1 items, send it to a bucket sorter

else{

// System.out.println(bucket + " " + index);

result = merge(result, bucketSort(bucket, index + 1));

}

}

System.out.println(result);

return result;

}

public static ArrayList merge(ArrayList arr1, ArrayList arr2){

arr1.addAll(arr2);

return arr1;

}

}

Source code online: https://github.com/JStephenD/Git/blob/master/DASTAL/src/PRELIM/BucketSort2.java