**6 kyu**

**Frequency Analysis With Buckets**

11389% of 5736 of178[ajLapid718](https://www.codewars.com/users/ajLapid718)

Java

* [TRAIN AGAIN](https://www.codewars.com/kata/frequency-analysis-with-buckets/train/java)
* [NEXT KATA](https://www.codewars.com/trainer/java)

Details

[Solutions](https://www.codewars.com/kata/frequency-analysis-with-buckets/solutions/java)

[Forks (1)](https://www.codewars.com/kata/frequency-analysis-with-buckets/forks/java)

[Discourse (21)](https://www.codewars.com/kata/frequency-analysis-with-buckets/discuss/java)

* Add to Collection
* |
* Share this kata:

Description:

Given an input array (arr) of positive integers, the objective is to return an output array where each index represents the amount of times an element appeared (frequency) in the input array.

More specifically, the element at each index of the output array will be an array (bucket) containing integers that appeared index-amount-of-times.

Otherwise, slot nulls (JavaScript, Java), None's (Python) nils (Ruby), or NULL's (C/C++) where appropriate. A valid array will always be provided.

If an array of [1,2,3,4,4,5,5,5] is passed in, the expected output should be: [null, [1,2,3], [4], [5], null, null, null, null, null].

Explanation:

// bucketize(arr) ======> outputArray

bucketize({1,2,3,4,4,5,5,5}) ======> {null, {1,2,3}, {4}, {5}, null, null, null, null, null}

An element cannot appear 0 times, so a null is placed at outputArray[0]. The elements 1, 2, and 3 appear once. This is why they are located at outputArray[1]. Notice the elements are grouped together in an array and sorted in ascending order. The element 4 appears twice. This is why it is located at outputArray[2]. The element 5 appears three times. This is why it is located at outputArray[3].

Although an integer could have possibly appeared four, five, six, seven, or eight times, this is not the case for this particular example. This is the reason why the elements at outputArray[4], outputArray[5], outputArray[6], outputArray[7], and outputArray[8] are all null values.

Examples:

bucketize({2,2,4,4,6,6,9,9,9,9}) ===> {null, null, {2,4,6}, null, {9}, null, null, null, null, null, null};

bucketize({3,3,3,3,2}) =============> {null, {2}, null, null, {3}, null};

bucketize({5,5,5,5,5}) =============> {null, null, null, null, null, {5}};

bucketize({77,3,40,40,40}) =========> {null, {3,77}, null, {40}, null, null};

bucketize({16,7,5,3,6,23}) =========> {null, {3,5,6,7,16,23}, null, null, null, null, null};

<https://www.codewars.com/kata/frequency-analysis-with-buckets/java>

*/\**

*\* To change this license header, choose License Headers in Project Properties.*

*\* To change this template file, choose Tools | Templates*

*\* and open the template in the editor.*

*\*/*

**package** javaapplication15;

**import** java.util.ArrayList;

**import** java.util.Collections;

**import** java.util.HashMap;

***/\*\****

***\****

***\* @author Usuario***

***\*/***

**public** **class** JavaApplication15 {

**public** **static** ArrayList<ArrayList<Integer>> bucketize(**final** **int**[] arr)

    {

        HashMap<Integer, Integer> hm  =

**new** HashMap();

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

**if**(hm.containsKey(arr[i])) {

               hm.put(arr[i], hm.get(arr[i]) + 1);

           } **else** {

               hm.put(arr[i], 1);

           }

       }

       ArrayList<ArrayList<Integer>> frec = **new** ArrayList();

*/\**

*for(int key : hm.keySet() ) {*

*frec[hm.get(key)] = new ArrayList();*

*}\*/*

**for**(**int** i =0; i<arr.length+1; i++) {

           frec.add(**new** ArrayList());

       }

**for**(**int** key : hm.keySet()) {

*//frec[hm.get(key)].add(key);*

           frec.get(hm.get(key)).add(key);

       }

**for**(**int** key : hm.keySet()) {

          Collections.sort(frec.get(hm.get(key)));

        }

        ArrayList<ArrayList<Integer>> ans =

**new** ArrayList();

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

**if**(frec.get(i).size() > 0) {

                ans.add(frec.get(i));

            } **else** {

                ans.add(**null**);

            }

        }

**return** ans;

    }

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

*// TODO code application logic here*

*//int[] arr =  {1,2,3,4,4,5,5,5};*

*//int[] arr = { 5, 5, 5, 9 };*

*//int[] arr = {2,2,4,4,6,6,9,9,9,9};*

*//{null, null, {2,4,6}, null, {9}, null, null, null, null, null, null};*

**int**[] arr = {1,2,3,4,4,5,5,5};

*//======> {null, {1,2,3}, {4}, {5}, null, null, null, null, null}*

        ArrayList<ArrayList<Integer>> res =  bucketize(arr);

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

**if**( res.get(i).size() > 0) {

               System.out.print("{");

**for**(**int** j =0; j<res.get(i).size(); j++) {

                    System.out.print(res.get(i).get(j) + " ");

                }

                System.out.print("}");

          } **else** {

               System.out.print("null ");

           }

*//System.out.println();*

      }

    }