

Sort elements by frequency | Set 1

Asked By Binod

Question:

Print the elements of an array in the decreasing frequency if 2 numbers have same frequency then print the one which came 1st

E.g. 2 5 2 8 5 6 8 8 output: 8 8 8 2 2 5 5 6.

METHOD 1 (Use Sorting)

- 1) Use a sorting algorithm to sort the elements $O(n \log n)$
- 2) Scan the sorted array and construct a 2D array of element and count $O(n)$.
- 3) Sort the 2D array according to count $O(n \log n)$.

Example:

Input 2 5 2 8 5 6 8 8

After sorting we get

2 2 5 5 6 8 8 8

Now construct the 2D array as

2, 2

5, 2

6, 1

8, 3

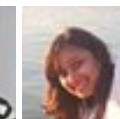
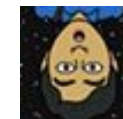
Google™ Custom Search



GeeksforGeeks



53,522 people like [GeeksforGeeks](#).



"LOG KHA KAHENSE"



[Interview Experiences](#)

[Advanced Data Structures](#)

[Dynamic Programming](#)

[Greedy Algorithms](#)

[Backtracking](#)

[Pattern Searching](#)

[Divide & Conquer](#)

[Mathematical Algorithms](#)

[Recursion](#)

Sort by count

8, 3
2, 2
5, 2
6, 1

There is one issue with above approach (thanks to ankit for pointing this out). If we modify the input to 5 2 2 8 5 6 8 8, then we should get 8 8 8 5 5 2 2 6 and not 8 8 8 2 2 5 5 6 as will be the case.

To handle this, we should use indexes in step 3, if two counts are same then we should first process(or print) the element with lower index. In step 1, we should store the indexes instead of elements.

Input 5 2 2 8 5 6 8 8

After sorting we get

Element 2 2 5 5 6 8 8 8

Index 1 2 0 4 5 3 6 7

Now construct the 2D array as

Index, Count

1, 2
0, 2
5, 1
3, 3

Sort by count (consider indexes in case of tie)

3, 3
0, 2
1, 2
5, 1

Print the elements using indexes in the above 2D array.



Popular Posts

[All permutations of a given string](#)

[Memory Layout of C Programs](#)

[Understanding "extern" keyword in C](#)

[Median of two sorted arrays](#)

[Tree traversal without recursion and without stack!](#)

[Structure Member Alignment, Padding and Data Packing](#)

[Intersection point of two Linked Lists](#)

[Lowest Common Ancestor in a BST.](#)

[Check if a binary tree is BST or not](#)

[Sorted Linked List to Balanced BST](#)

METHOD 2(Use BST and Sorting)

1. Insert elements in BST one by one and if an element is already present then increment the count of the node. Node of the Binary Search Tree (used in this approach) will be as follows.

```
struct tree
{
    int element;
    int first_index /*To handle ties in counts*/
    int count;
}BST;
```

2.Store the first indexes and corresponding counts of BST in a 2D array.

3 Sort the 2D array according to counts (and use indexes in case of tie).

Time Complexity: $O(n \log n)$ if a Self Balancing Binary Search Tree is used.

METHOD 3(Use Hashing and Sorting)

Using a hashing mechanism, we can store the elements (also first index) and their counts in a hash. Finally, sort the hash elements according to their counts.

These are just our thoughts about solving the problem and may not be the optimal way of solving. We are open for better solutions.

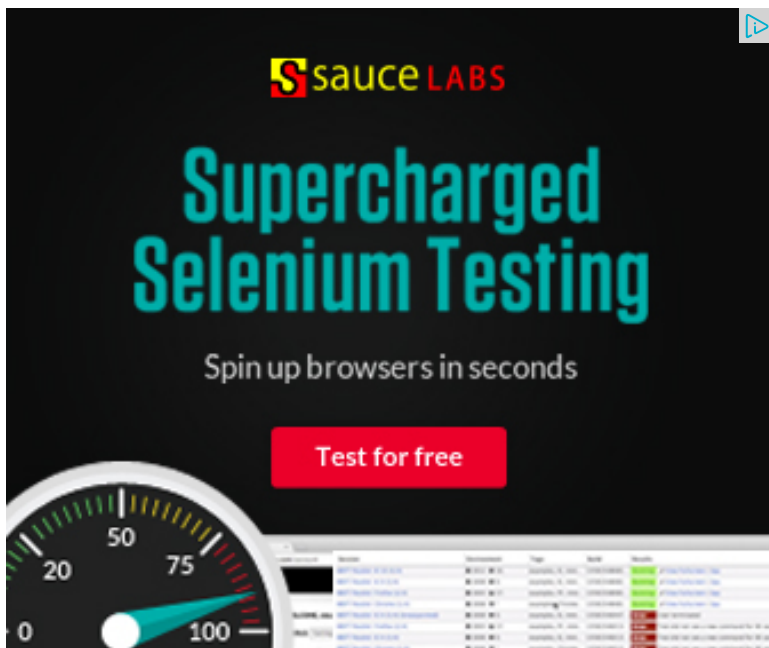
Related Links

<http://www.trunix.org/programlama/c/kandr2/krx604.html>

<http://drhanson.s3.amazonaws.com/storage/documents/common.pdf>

<http://www.cc.gatech.edu/classes/semantics/misc/pp2.pdf>





the open cloud company

705



Subscribe

Recent Comments

Aman Hi, Why arent we checking for conditions...

Write a C program to Delete a Tree. · 22 minutes ago

kzs please provide solution for the problem...

Backtracking | Set 2 (Rat in a Maze) · 25 minutes ago

Sanjay Agarwal bool

tree::Root_to_leaf_path_given_sum(tree...

Root to leaf path sum equal to a given number · 50 minutes ago

GOPI GOPINATH @admin Highlight this sentence "We can easily...

Count trailing zeroes in factorial of a number · 52 minutes ago

newCoder3006 If the array contains negative numbers also. We...

Find subarray with given sum · 1 hour ago

newCoder3006 Code without using while loop. We can do it...

Find subarray with given sum · 1 hour ago

Related Topics:

- Remove minimum elements from either side such that $2 \times \text{min}$ becomes more than max
- Divide and Conquer | Set 6 (Search in a Row-wise and Column-wise Sorted 2D Array)
- Bucket Sort
- Kth smallest element in a row-wise and column-wise sorted 2D array | Set 1
- Find the number of zeroes
- Find if there is a subarray with 0 sum
- Divide and Conquer | Set 5 (Strassen's Matrix Multiplication)
- Count all possible groups of size 2 or 3 that have sum as multiple of 3



9



Tweet

0



0

Writing code in comment? Please use ideone.com and share the link here.

44 Comments

GeeksforGeeks

Sort by Newest ▾





JOIN THE DISCUSSION...

AdChoices

[▶ Java Sort](#)

[▶ Sort Java](#)

[▶ How to Sort Java](#)

AdChoices

[▶ Sort Java](#)

[▶ How to Sort Java](#)



zzerr • 22 days ago

```
#include<iostream>
```

```
#include<string>
```

```
#include<time.h>
```

```
#include<algorithm>
```

```
#include<unordered_map>
```

```
#include<map>
```

```
using namespace std;
```

```
class Elem
```

```
{
```

```
public:
```

```
Elem(){}
```

[see more](#)

[^](#) | [v](#) • [Reply](#) • [Share](#) ›



Mari • 2 months ago

In third Solution, Step 2, Since the frequency of any number lies between 1 to 10000, the total frequency of all numbers over all running time of this solution will be $O(n)$ (Hash table operations + counting).

[^](#) | [v](#) • [Reply](#) • [Share](#) ›



Meenal • 4 months ago

I tried to modify merge sort for storing original indexes after modification it is not working.

where I have gone wrong Please help me find the issue..

Please find my code below.

<http://ideone.com/ORQLql>

^ | v • Reply • Share ›



Coder011 → Meenal • 4 months ago

You havent mentioned what does the ordered pair $\{x,y\}$, in your array 'i' array value and, x as index. But even going by that logic where have you
Link to Ideone: <http://ideone.com/Y6GeeA>

^ | v • Reply • Share ›



Meenal → Coder011 • 4 months ago

Hi, your assumption of $\{x,y\}$ is correct.

This part of code is written just for sorting the array and storing elements in `a[i][1]` only. As it is not sorting correctly itself. I could

I am not able to get why making it a 2D array affected it, as with merge sort program works perfectly fine.

^ | v • Reply • Share ›



Coder011 → Meenal • 4 months ago

In the stdout of the program (taken from Ideone):

553456789102

Array after sorting:

Index after sorting:

9 1 2 3 4 5 6 7 8 0

Array after sorting:

2 3 4 5 6 7 8 9 10 55

I suppose, the array has been sorted. Have you tried it c

Otherwise the code looks fine to me.

^ | v · Reply · Share ›



Meenal → Coder011 · 4 months ago

Thanks for checking it..

I was facing problem with same test case, there was or instead of $(L[i][1]$

$< R[j][1])$ i used $(L[i] < R[j])$ by mistake... Gues you have

Thanks

^ | v · Reply · Share ›



Guest · 4 months ago

<http://ideone.com/Eplnhc>

^ | v · Reply · Share ›



jedi · 7 months ago

```
#include<stdio.h>
```

```
//given an array, first sort in nlogn time using merge sort
```

```
mergesort(int A[][2],int left,int right)
```

```
{
```

```
if(left>=right)
```

```
return A[left][1];
```

```
int mid=(left+right)/2;
```

```
mergesort(A,left,mid);
```

```
mergesort(A,mid+1,right);
```

```
merge(A,left,mid,right);
```

```
}
```

[see more](#)

^ | v • Reply • Share ›



jedi → jedi • 7 months ago

sorry about that..

^ | v • Reply • Share ›



Suryabhan Singh • 8 months ago

How i m going to use hashing if range is unknown ????

^ | v • Reply • Share ›



alien → Suryabhan Singh • 8 months ago

You need not use the same element to store in hashing array. You can element and put them into a specific range and then put into hash table

^ | v • Reply • Share ›



Cracking The Code • 11 months ago

How about this.

```
import java.util.ArrayList;
import java.util.Collections;
import java.util.Comparator;
import java.util.HashMap;
import java.util.List;
import java.util.Map;
```

```
class SortByFrequency {
```

```
class FreqIndexPairs {
```



```
public Integer Frequency ;.
```

```
public Integer FirstIndex ;.
```

```
public FreqIndexPairs() {.
```

```
Frequency=0;.
```

[see more](#)

^ | v • Reply • Share ›



Parish Jain • 11 months ago

:0

^ | v • Reply • Share ›



Varun Jain • 11 months ago

I suppose we could modify second method by making a heap based on count
extract element one-by-one.

^ | v • Reply • Share ›



alien → Varun Jain • 8 months ago

it would be in $O(n)$ time but the question is you need to change the key

^ | v • Reply • Share ›



hbandi • a year ago

see this

```
import java.util.*;
```

```
public class NumberCount {
```

```

public static void main(String[] args) {
    NumberCount nc = new NumberCount();
    int[] array = { 2, 5, 2, 8, 5, 6, 8, 8 };
    System.out.println("Map is :: " + nc.getMapForCounts(i
}

private Map sortedMap(Map map) {
    List<Integer> list = new ArrayList(map.entrySet());
    Collections.sort(list, new Comparator() {
        public int compare(Object one, Object two) {

```

[see more](#)

^ | v • Reply • Share ›



akshat gupta • a year ago

GeeksforGeeks?-in Method 2:

BST can have height Ranging from $\lg 10$ - 10,
so $T(n) = O(10 \cdot n)$ or $O(n)$,
not $O(n \lg n)$

^ | v • Reply • Share ›



GeeksforGeeks • a year ago

Abhinav, counting sort doesn't work for random input. It works only when

1 ^ | v • Reply • Share ›



Guest → GeeksforGeeks • 15 days ago

@GeeksforGeeks, We can use the idea similar to counting sort and $O(n)$ space. Although we have random input, but the frequency of any element is between 1 to n , and we can exploit this property.

^ | v • Reply • Share ›



use count sort.. as simple as that.

^ | v • Reply • Share ›



Ankit Malhotra • a year ago

$O(n^2)$ code follows. First build a linked list with frequency and lowest index. T
The case of equal frequencies is handled easily here as the linked list anyway r

```
#include <iostream>
#define MAXCOUNT 1000
using namespace::std;
typedef int element;
typedef int counter;

struct datanode
{
    element data;
    counter frequency;
    counter index;
    datanode * next;
    datanode (element data, counter index)
    {
        this->data = data;
```

[see more](#)

^ | v • Reply • Share ›



Ankit Malhotra → Ankit Malhotra • a year ago

Added code to destroy binary tree.

```
void destroy (ptr p)
{
    if (!p) return;
```

```

ptr k = p;
destroy (p->left);
destroy (p->right);
delete k;
}

int main()
{
    element term[] = {-10, 0, -30, 0, -30, 0, -36, -10, -25}, rec
    counter count = sizeof(term)/sizeof(element), index[MAXCOUNT]
    cout << "Original Array ..." << endl;
    for (counter i = 0; i < count; i++) cout << term[i] << ' ';
    cout << endl;

```

[see more](#)

^ | v • Reply • Share ›



ankit • 2 years ago

In my opinion maxHeap will be the best ds, heapify based on the count, and st node. If there is a tie in the frequency, we have to just check the children of the

^ | v • Reply • Share ›



alien → ankit • 8 months ago

it would be in $O(n \log n)$. If linear time algorithm is there, we should use it

^ | v • Reply • Share ›



algobard • 2 years ago

Can you guys post the codes for this question? Please!

^ | v • Reply • Share ›



saurabh • 2 years ago

It can be done by performing Counting Sort twice.

For the first time the frequency of each number **is** counted **in** another
For the second time, we perform counting on the frequencies with same
Space complexity = $O(4n)$. Time Complexity = $O(5n)$. So basically both :

```
#include <stdio.h>
# define ARY_SIZE 50
//a[] is input array, d[] is output array
int sortByFreq(int a[], int d[], int len) {
    int b[len], c[len], i, newLen = 0;
    for(i = 0; i < len; i++) {
        b[i] = 0;
        c[i] = 0;
    }
    // Frequency of each element is stored in b[]
    for(i = 0; i < len; i++)
        b[a[i]]++;
```

[see more](#)

^ | v • Reply • Share ›



Ankit Malhotra → saurabh • a year ago

This code gives segmentation fault on the simplest of inputs. Also has inability to use negative numbers at all as terms are used as direct array indices.

^ | v • Reply • Share ›



student • 3 years ago

can't we use array of structures instead of using bst ?

^ | v • Reply • Share ›



rajcools • 3 years ago

//Code for method 1

```
//anybody who wants to test using different values please change //va:
//MergeSort(initial2D,0,7); here 7 is sizeof array -1
//CreateArray(initial2D,8); 8 is sizeof array
//MergeSort1(initial2D,0,3); 3 is number of different values in //arra

void swap(int &a,int &b)
{
    int temp;
    temp = b;
    b=a;
    a=temp;
}
void Merge(int arr[][3], int p, int q, int r)
{
    int n1 = q-p+1;
    int n2 = r-q;
```

[see more](#)

^ | v • Reply • Share ›



rajcools • 3 years ago

I have written the code for method 1.

I have created a 2-D array that stores - value, index and count
step1

- 1) sort this 2-D array on the basis of number
- 2) count the number of times value exist
- 3) sort on the basis of count taking index into consideration when there is a tie
- 4)output the value

My implementation

- 1)merge sort used for this function name MergeSort O(n)

2)in 2d array calculate count for every value and store in 1st entry corresponding to this value count value is zero.And then delete repetitive entries value is non zero remain. 2d array is condensed. If there are four different values useful

This is in function create array. $O(n)$

3)this is accomplished again via mergesort(different from mergesort of 1st step)

4)this is done in main

^ | v • Reply • Share ›



abc • 3 years ago

A small correction:

In the first example the treap I drew at the end is wrong. It should be:

```
      8 (c=3, fi =4)
     /
    2 (c=2, fi =0)
   \
    5 (c=2, fi=1)
```

^ | v • Reply • Share ›



abc • 3 years ago

Just thinking aloud, I somewhere feel we can use a treap here? This looks like it can be used.

Say the input is:

2,5,2,5,8,8,8

o/p should be: 8,8,8,2,2,5,5

So here count will serve as priority for a treap.

So let's enter numbers.

(i am using 'c' for count and 'fi' as the first index in the brackets for a node)

```

                2 (c=1, fi=0)
2) 5
                2 (c=1, fi=0)
            / \
            5
        (c=1, fi=1)
3) enter 2. So here as its already present we increment the count
                2 (c=2, fi=0)

```

[see more](#)

^ | v • Reply • Share ›



Vamshi Krishna • 3 years ago

In Method 1 using [index, count] approach - there is no need to sort the array to traverse thru the input array, add new index in the if the element is new, else increment entry.

This can be done in $O(n)$ time complexity, where as sorting would take $O(n \log n)$

^ | v • Reply • Share ›



WgpShashank • 3 years ago

Hi Sandeep, GeeksforGeeks

i am just about to finish the exact working solution my program time complexity is $O(n)$ can we do it in $O(1)$ maybe i am missing something..given suggestion....please..so reply fast...what i am trying to say is that once we get the frequency of each element in some array or hashset or simple set ..so space is $O(n)$..so can we do better?

Reply ASAP

Shashank

^ | v • Reply • Share ›



Shashank Mani Narayan • 3 years ago



<http://codepad.org/JzSNCEkL>

will help just little modification needed

^ | v • Reply • Share ›



Rini • 3 years ago

In the hashing technique, if we modify it to maintain (instead of the count) a link element in the array, or say the index in the array, then sorting would no longer ensure it's stable too.

^ | v • Reply • Share ›



Sam • 4 years ago

My solution is as following and appreciate your code optimization

```
public class FreqIndexPair
{
    public int Freq { get; set; }
    public int FirstIndex { get; set; }
}

public static int[] SortByFrequencyAndFirstOccurence(int[] array)
{
    Dictionary hash = new Dictionary();
    for (int i = 0; i < array.Length; i++)
    {
        if (!hash.ContainsKey(array[i]))
        {
            hash.Add(array[i], new FreqIndexPair() { FirstIndex = i, Freq = 1 });
        }
        else
        {
            hash[array[i]].Freq++;
        }
    }
}
```

[see more](#)

^ | v • Reply • Share ›



Sandeep → Sam • 4 years ago

@Sam: Thanks for sharing the code. Could you also write the algorithm please?

^ | v • Reply • Share ›



GeeksforGeeks → Sandeep • 3 years ago

@Algoseekar & @rahul:

We will try to add code to this post. In the mean time, you can find implementation.

<http://www.trunix.org/programl...>

<http://drhanson.s3.amazonaws.com/storage/documents/comr>

<http://www.cc.gatech.edu/class...>

1 ^ | v • Reply • Share ›



rahul → Sandeep • 3 years ago

@sandeep can you please post the exact working code for this anywhere ..??

^ | v • Reply • Share ›



Algoseekar → Sandeep • 3 years ago

@sandeep...Can You Provide The Solution for This Question U

^ | v • Reply • Share ›



geeksforgeeks • 4 years ago

@ankit: Thanks for reporting this case. We have modified the above methods

^ | v • Reply • Share ›



ankit • 4 years ago



If we modify the input as: 0 2 0 0 0 0, the code should print 0 0 0 0 2 0, in case.

^ | v • Reply • Share ›



Subscribe



Add Disqus to your site

