All Tracks › Algorithms › Sorting › Counting Sort

Algorithms

3 LIVE EVENTS

Solve any problem to achieve a rank

View Leaderboard

Topics: Counting Sort

Counting Sort

TUTORIAL PROBLEMS

In Counting sort, the frequencies of distinct elements of the array to be sorted is counted and stored in an auxiliary array, by mapping its value as an index of the auxiliary array.

Algorithm:

Let's assume that, array A of size N needs to be sorted.

- Initialize the auxillary array $Aux[]$ as 0.
  Note: The size of this array should be $\geq max(A[])$.
- Traverse array A and store the count of occurrence of each element in the appropriate index of the Aux array, which means, execute `Aux[A[i]]++` for each i, where i ranges from $[0, N-1]$.
- Initialize the empty array $sortedA[]$
- Traverse array Aux and copy i into $sortedA$ for $Aux[i]$ number of times where $0 \leq i \leq max(A[])$.

Note: The array A can be sorted by using this algorithm only if the maximum value in array A is less than the maximum size of the array Aux. Usually, it is possible to allocate memory up to the order of a million (10^6). If the maximum value of A exceeds the maximum memory- allocation size, it is recommended that you do not use this algorithm. Use either the quick sort or merge sort algorithm.

Implementation:

Assume that the maximum element that can be in the array is K .
Now take an $Aux[]$ array of size $K+1$.
$A[]$ = Array to be sorted.
$sortedA[]$ = Sorted version of $A[]$.

```
void counting_sort(int A[], int Aux[], int sortedA[], int N) {

    // First, find the maximum value in A[]
    int K = 0;
    for(int i=0; i<N; i++) {
        K = max(K, A[i]);
    }

    // Initialize the elements of Aux[] with 0
    for(int i=0 ; i<=K; i++) {
        Aux[i] = 0;
    }

    // Store the frequencies of each distinct element of A[],
    // by mapping its value as the index of Aux[] array
    for(int i=0; i<N; i++) {
        Aux[A[i]]++;
    }

    int j = 0;
    for(int i=0; i<=K; i++) {
        int tmp = Aux[i];
        // Aux stores which element occurs how many times,
        // Add i in sortedA[] according to the number of times i occured in
A[]
        while(tmp--) {
            //cout << Aux[i] << endl;
            sortedA[j] = i;
            j++;
        }
    }
}
```

Example:
Say $A = \{5, 2, 9, 5, 2, 3, 5\}$.

Aux will be of the size $9+1$ i.e. 10

$Aux = \{0, 0, 2, 1, 0, 3, 0, 0, 0, 2\}$.
Notice that $Aux[2] = 2$ which represents the number of occurrences of 2 in $A[]$. Similarly $Aux[5] = 3$ which represents the number occurrences of 5 in $A[]$.

After applying the counting sort algorithm, $sortedA[]$ will be $\{2, 2, 3, 5, 5, 5, 9\}$

Time Complexity:
The array A is traversed in $O(N)$ time and the resulting sorted array is also computed in $O(N)$ time. $Aux[]$ is traversed in $O(K)$ time. Therefore, the overall time complexity of counting sort algorithm is $O(N + K)$.

Contributed by: Ravi Ojha

TEST YOUR UNDERSTANDING

Counting Sort

You have been given an integer array *A* of size *N*. Each element of the array ranges between *1* and 10^5. You need to find the frequency of each distinct element of the array. The elements need to be present in the output in ascending order. You need to print the value and then frequency of each distinct element.

Input Format:

The first line contains a single integer *N* denoting the size of the array. The next line contains *N* space separated integers, denoting the elements of the array.

Output Format

For each distinct integer, print its value and then frequency in a new line. The distinct integers should appear in the output in ascending order.

Constraints

$1 \leq N \leq 100$

$1 \leq A[i] \leq 100$

SAMPLE INPUT

```
5
5 4 3 2 1
```

SAMPLE OUTPUT

```
1 1
2 1
3 1
4 1
5 1
```

Enter your code or Upload your code as file. Save C (gcc 5.4.0) ▾

```
/*
// Sample code to perform I/O:
#include <stdio.h>

int main(){
    int num;
    scanf("%d", &num);                          // Reading input from STDIN
    printf("Input number is %d.\n", num);       // Writing output to STDOUT
}

// Warning: Printing unwanted or ill-formatted data to output will cause the test cases t
*/

// Write your code here

```

1:1

Provide custom input

COMPILE & TEST SUBMIT

COMMENTS (22)

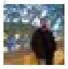
SORT BY: Relevance▾

Login/Signup to Comment

Vinh Thomas a year ago

"Each element of the array ranges between 1 and 10^5" and why the contrains 1<=A[i]<=100?

▲ 8 votes • Reply • Message • Permalink



defilerc 7 months ago

The correct constraint is 1<=A[i]<=100.

▲ 1 vote • Reply • Message • Permalink

shubham gupta 2 years ago

Please add a precondition that input array should have positive integers else we have to extend this count sort with some processing .

▲ 3 votes ● Reply ● Message ● Permalink

ADRISH GHOSH a year ago

Last element of Aux[] will be 1 in the given example as there is only one '9' in the array A[].

▲ 1 vote ● Reply ● Message ● Permalink

humblefool a year ago

```
#include <iostream>
using namespace std;
int main()
{
int n,t,i,a[100001]={0};
std::cin>>n;
for(i=0;i<n;i++)
{ std::cin>>t;
a[t]++;
}
for(i=0;i<=100000;i++){
if(a[i]>0)
std::cout<<i<<" "<<a[i]<<endl;
}
}
```

▲ 1 vote ● Reply ● Message ● Permalink

Pankaj Bhushan a year ago

I am having some trouble in passing test cases.
For some reason the same code produces correct output on my local machine but is different when I submit here (missing repetitive values). Also as per correct output shown for test#1, it has repetitive values printed along with their frequency whereas for test#2 only distinct values are printed with corresponding frequency. How are we suppose to produce two different result types for different cases?

▲ 0 votes ● Reply ● Message ● Permalink

Pankaj Bhushan Edited a year ago

```
import java.util.Scanner;
class TestClass {
public static void main(String[] args) {
Scanner s = new Scanner(System.in);
int N = s.nextInt();
int A[] = new int[N];
for (int i = 0; i < N; i++) {
A[i] = s.nextInt();
}
int sortedA[] = new int[N];
int Aux[] = null;
// Perform counting sort
countingSort(A, Aux, sortedA);
s.close();
}
private static void countingSort(int[] A, int[] Aux, int[] sortedA) {
int k = 0;
// Find the maximum value
for (int i = 0; i < A.length; i++)
k = Math.max(A[i], k);
Aux = new int[k + 1];
for (int i = 0; i < A.length; i++)
Aux[A[i]]++;
int j = 0;
for (int i = 0; i < Aux.length; i++) {
int tempCount = Aux[i];
while (tempCount!= 0) {
sortedA[j++] = i;
tempCount--;
}
}

//Printing the desired output
for (int i = 0; i < sortedA.length; i++) {
System.out.println(sortedA[i] + " " + Aux[sortedA[i]]);
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Raghav Ravi Prakash a year ago

Java solution

```
import java.util.*;
public class CountingSort
{
public static void main(String[] args)
{
Scanner in = new Scanner(System.in);
int N = in.nextInt();

Integer[] array = new Integer[N+1];
array[0] = 0;
for(int i = 1; i < array.length; i++)
array[i] = in.nextInt();

int max = (int) Collections.max(Arrays.asList(array));
int[] countArray = new int[max+1];
Arrays.fill(countArray,0);

countSort(array,countArray);
}

private static void countSort(Integer[] array, int[] countArray)
{
for(int i = 1; i < array.length; i++)
countArray[array[i]]++;

for(int i = 1; i < countArray.length; i++)
{
int key = i;
int freq = countArray[i];

if(freq != 0)
printOutput(key,freq);
}
}

public static void printOutput(int key, int freq)
{
System.out.println(key + " " + freq);
}
}
```

▲ 0 votes • Reply • Message • Permalink

Vikas Gautam a year ago

```
#include <stdio.h>
int main()
{
int *A,N,i,j;
scanf("%d",&N);
A = (int *)malloc(N*sizeof(int));
for(i=0;i<N;i++)
{
scanf("%d",&A[i]);
}

int max = A[0];

for(i=1;i<N;i++)
{
if(max<A[i])
{
max = A[i];
}
}

int Aux[max+1];

for(i=0;i<=max;i++)
{
Aux[i] = 0;
}

for(i=0;i<N;i++)
{
Aux[A[i]]++;
}

int sortedA[N];
j=0;
for(i=0;i<=max;i++)
{
int temp = Aux[i];

while(temp--)
{
sortedA[j]=i;
j++;
}
}

for(i=0;i<N;i++)
{
int temp = Aux[sortedA[i]];
while(temp--)
{
printf("%d %d\n",sortedA[i],temp+1);
temp = 0;
Aux[sortedA[i]]=0;
}
}

return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Akash Chandra a year ago

```
a=[sorted(map(int,input().split())) for _ in range(2)][1]
[print("{} {}".format(i,a.count(i))) for i in set(a)]
```

IN PYTHON 3.5

▲ 0 votes ● Reply ● Message ● Permalink

Vivek Shah a year ago

C++ Solution :

```
#include <bits/stdc++.h>
#define ll long long int
#define rep(i,a,b) for(ll i=a;i<b;i++)
using namespace std;
int main()
{
ll n;
cin>>n;
ll a[n];
ll k = 0;
rep(i,0,n){
cin>>a[i];
k = max(k,a[i]);
}
ll b[k+1] = {0};
rep(i,0,n)b[a[i]]++;
rep(i,0,k+1)if(b[i]!=0)cout<<i<<" "<<b[i]<<"\n";
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

basantlalnit 10 months ago

Solution in C passes all test case

```
#include <stdio.h>
int max(int k,int x)
{
if(k<x)
{
return x;
}
else
{
return k;
}

}
void count_sort(int arr[],int n)
{
int k=0;
for(int i=0;i<n;i++)
{
k=max(k,arr[i]);
}
int Aux[k+1];
for(int i=0;i<k+1;i++)
{
Aux[i]=0;
}
for(int j=0;j<n;j++)
{
Aux[arr[j]]++;
}
int arr1[n];
int l=0;
for(int i=0;i<k+1;i++)
{
int temp=Aux[i];
while(temp--)
{
arr1[l++]=temp;
}
}
for(int i=0;i<=k;i++)
{
if(Aux[i]==0)
{
continue;
}
else
{
printf("%d %d\n",i,Aux[i]);
}
}
}
int main()
{
int n;
scanf("%d",&n);
int arr[n],i;
for(i=0;i<n;i++)
{
scanf("%d",&arr[i]);
}
count_sort(arr,n);
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

El Houcine Arg 10 months ago

My solution :

```
import java.util.*;
class TestClass {
public static HashSet counting_sort(int[] A,int[] Aux,int n,int max){

int j=0;
HashSet<Integer> sorted=new HashSet<Integer>();
for(int i=0;i<=max;i++){
int emp=Aux[i];
if(emp>0){
sorted.add(i);
}

}
return sorted;
}
public static void main(String[] args) {
Scanner s=new Scanner(System.in);
int n=s.nextInt();
int[] A=new int[n];
int max=0;
for(int i=0;i<n;i++){
int m=s.nextInt();
if(m>max){
max=m;
}
A[i]=m;
}
int[] Aux=new int[max+1];
for(int i=0 ; i<=max; i++) {
Aux[i] = 0;
}
for(int i=0; i<n; i++) {
Aux[A[i]]++;
}
HashSet<Integer> sort=new HashSet<Integer>();
sort=counting_sort(A,Aux,n,max);
for(int b:sort){
System.out.println(b+" "+Aux[b]);
}

}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Dird 9 months ago

This counting sort example code is not good because I don't think it can be used with Radix Sort on elements larger than 10

▲ 0 votes ● Reply ● Message ● Permalink

Ayush Gupta 8 months ago

correct code in c++

```
#include<bits/stdc++.h>
using namespace std;
int main(){
int n;
cin>>n;
int a[n],i;
for(i=0;i<n;i++){
cin>>a[i];
}
int max=a[0];
for(i=1;i<n;i++)
if(max<a[i])
max=a[i];
int arr[max+1]={0};
for(i=0;i<n;i++){
arr[a[i]]++;
}
for(i=0;i<=max;i++){
if(arr[i]){
cout<<i<<" "<<arr[i]<<endl;
}
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Abhishek Ranjan 7 months ago

```
int main()
{
int n,max;
int a[100000];
cin>>n;
for(int i=0;i<n;i++)
{
cin>>a[i];
}
sort(a,a+n);
max = a[n-1];
int hash[max+1] = {0};
for(int i=0;i<n;i++)
{
hash[a[i]]++;
}
for(int i=1;i<max+1;i++)
{
if(hash[i]>0)
cout<<i<<" "<<hash[i]<<endl;
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Sirisha K 6 months ago

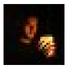
```
n=int(input())
a=input()
a=[int(x) for x in a.split(' ')]
my_dict ={i:a.count(i) for i in a}
for i,v in my_dict.items():
print(i,v)
```

▲ 0 votes ● Reply ● Message ● Permalink

Abdullah Dogar 6 months ago

Can we apply counting sort on characters and floats??????

▲ 0 votes ● Reply ● Message ● Permalink



XandriethXs 6 months ago

|| Java Solution ||

```
import java.util.*;
class TestClass {
public static void main(String args[] ) throws Exception {
Scanner s = new Scanner(System.in);
int N = s.nextInt();
int A[] = new int[N];
int max = 0;
for (int i=0;i<N;i++)
{
A[i] = s.nextInt();
if (A[i]>max)
{
max = A[i];
}
}
int Ac[] = new int[max+1];
for (int i=0;i<N;i++)
{
Ac[A[i]]++;
}
for (int i=0;i<=max;i++)
{
if(Ac[i]>0)
{
System.out.println(i + " " + Ac[i]);
}
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Mayank Shekhar 2 months ago

Simple hashing technique:

```
#include<bits/stdc++.h>
using namespace std;
#define MAX 100005
int main()
{
int n;
cin>>n;
int arr[n];
int i;
for(i=0;i<n;i++)
cin>>arr[i];
int count[MAX]={0};
for(i=0;i<n;i++)
count[arr[i]]++;
for(i=0;i<MAX;i++)
{
if(count[i])
cout<<i<<" "<<count[i]<<endl;
}
return 0;
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Baris Gebesoglu 2 months ago

```
import java.io.BufferedReader;
import java.io.InputStreamReader;
//import for Scanner and other utility classes
import java.util.*;
// Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail
class TestClass {
public static void main(String args[] ) throws Exception {
/* Sample code to perform I/O:
* Use either of these methods for input
//BufferedReader
BufferedReader br = new BufferedReader(new InputStreamReader(System.in));
String name = br.readLine(); // Reading input from STDIN
System.out.println("Hi, " + name + "."); // Writing output to STDOUT
//Scanner
Scanner s = new Scanner(System.in);
String name = s.nextLine(); // Reading input from STDIN
System.out.println("Hi, " + name + "."); // Writing output to STDOUT
*/
// Write your code here

Scanner s = new Scanner(System.in);
int arrSize = new Integer(s.nextLine());
String sr = s.nextLine();
int [] inputArray = new int [arrSize];
String [] strArray = sr.split(" ");
for(int i=0;i<strArray.length;i++) {
inputArray[i] = new Integer(strArray[i]);
}

Map<Integer,Integer> arrMap = new HashMap<Integer,Integer>();
int highest = 0;

for(int i=0;i<inputArray.length;i++) {
if(inputArray[i]>highest) highest = inputArray[i];
if(arrMap.get(inputArray[i])!=null)
arrMap.put(inputArray[i],(arrMap.get((inputArray[i])) + 1));
else
arrMap.put(inputArray[i],1);
}

//int [] sortArr = new int [999999];

for(int i=0;i<highest+1;i++) {
if(arrMap.get(i)!=null)
System.out.println(i + " " + arrMap.get(i));
else
continue;
}


}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

Partha Sarathi Panda a month ago

Simple Java Solution

```
import java.util.Scanner;
public class CountingSortProg {
public static void main(String[] args) {
Scanner sc = new Scanner(System.in);
int n = sc.nextInt();
int aux[] = new int[100001];
for(int i=0;i<n;i++){
aux[sc.nextInt()]++;
}

for(int i=0;i<aux.length;i++){
if(aux[i]>0)
System.out.print(i +" "+ aux[i] +"\n");
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

About Us Innovation Management Technical Recruitment
University Program Developers Wiki Blog
Press Careers Reach Us

?