



Signup and get free access to 100+ Tutorials and Practice Problems

[Start Now](#)

3

LIVE EVENTS

[All Tracks](#) > [Algorithms](#) > [Sorting](#) > [Bucket Sort](#)

Algorithms

🔔 Solve any problem to achieve a rank

[View Leaderboard](#)Topics:

Bucket Sort

[TUTORIAL](#) [PROBLEMS](#)

What is Bucket Sort ?

Bucket sort is a comparison sort algorithm that operates on elements by dividing them into different buckets and then sorting these buckets individually. Each bucket is sorted individually using a separate sorting algorithm or by applying the bucket sort algorithm recursively. Bucket sort is mainly useful when the input is uniformly distributed over a range.

Assume one has the following problem in front of them:

One has been given a large array of floating point integers lying uniformly between the lower and upper bound. This array now needs to be sorted. A simple way to solve this problem would be to use another sorting algorithm such as Merge sort, Heap Sort or Quick Sort. However, these algorithms guarantee a best case time complexity of $O(N \log N)$. However, using bucket sort, the above task can be completed in $O(N)$ time. Let's have a closer look at it.

Consider one needs to create an array of lists, i.e of buckets. Elements now need to be inserted into these buckets on the basis of their properties. Each of these buckets can then be sorted individually using Insertion Sort. Consider the pseudo code to do so:

```
void bucketSort(float[] a,int n)
{
    for(each floating integer 'x' in n)
    {
```

?

```
        insert x into bucket[n*x];
    }
    for(each bucket)
    {
        sort(bucket);
    }
}
```

Time Complexity:

If one assumes that insertion in a bucket takes $O(1)$ time, then steps 1 and 2 of the above algorithm clearly take $O(n)$ time.

Contributed by: Anand Jaisingh

Did you find this tutorial helpful?



YES



NO

TEST YOUR UNDERSTANDING

Bucket Sort

You have been given an array A of size N . The array contains integers. You need to divide the elements of this array into buckets on the basis of the number of set bits in its binary representation. You need to then print the content of each bucket in a new line. The buckets should appear in the output in ascending order, i.e the bucket that stands for lesser number of set bits should appear before any other bucket which stands for higher number of set bits. The elements of each bucket should appear in ascending order too. That is if 2 integers appear in the same bucket, the one with the lower value should appear in the bucket list before the one with higher value.

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:

The output should contain the number of lines equal to the number of distinct bucket. If a bucket remains empty, it should not appear in the output. Print the contents of each bucket on a new line.

Constraints:

$$1 \leq N \leq 10^3$$

?

$$1 \leq A[i] \leq 10^3$$

Note

It is guaranteed that each array element is unique.

SAMPLE INPUT

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

SAMPLE OUTPUT

```
1 2 4
3 5
```

Enter your code or [Upload your code](#) as file.

Save

C (gcc 5.4.0)



```
1  /*
2  // Sample code to perform I/O:
3  #include <stdio.h>
4
5  int main(){
6      int num;
7      scanf("%d", &num);           // Reading input from STDIN
8      printf("Input number is %d.\n", num); // Writing output to STDOUT
9  }
10
11 // Warning: Printing unwanted or ill-formatted data to output will cause the test cases to fail
12 */
13
14 // Write your code here
15
```

1:1

☒ Provide custom input

COMPILE & TEST

SUBMIT

COMMENTS (11)

SORT BY: Relevance

Login/Signup to Comment



Nguyen Hai Trieu Edited 2 years ago

here is my solution:

```
#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
typedef vector< vector<int> > vvi;
int count(int k)
{
    int dem = 0;
    for (int i = 0; i<10; i++)
    {
        if (k & 1 << i)
            dem++;
    }
    return dem;
}
void bucketSort(int a[], int n, vvi b)
{
    for (int i = 0; i<n; i++)
    {
        b[count(a[i])].push_back(a[i]);
    }
    for (int i = 0; i<10; i++)
    {
        sort(b[i].begin(), b[i].end());
    }
    for (int i = 0; i<10; i++)
    {
        for (int j = 0; j<b[i].size(); j++)
        {
            cout << b[i][j];
            if (j < b[i].size() - 1)
                cout << " ";
        }
        if (i < n - 1)
            cout << endl;
    }
}
int main()
{
    int n;
    int a[1010];
    vvi b;
    b.resize(11);
    cin >> n;
    for (int i = 0; i<n; i++)
        cin >> a[i];
    bucketSort(a, n, b);
    return 0;
}
```

▲ 1 vote ● Reply ● Message ● Permalink



RP a year ago

Do not post solutions in comments. Hints are fine. Doubts too. Plain solution just spoils the question.

?

▲ 4 votes ● Reply ● Message ● Permalink

**Shubham Kumar** 5 months ago

solution:

```

#include <bits/stdc++.h>
#include <iomanip>
using namespace std;
#define ll long long
#define ull unsigned long long
#define opt ios_base::sync_with_stdio(false);cin.tie(NULL);cout.tie(NULL);
#define pb push_back
#define mp make_pair
#define rep(i,n) for(i=0;i<n;i++)
#define repl(i,a,b) for(i=a;i<b;i++)
#define repa(i,a,b) for(i=a;i<=b;i++)
#define repr(i,n) for(i=n-1;i>=0;i--)
#define nl cout<<"\n";
#define M 1000000007
#define N 1000005

ll nosb(ll x)
{
    ll cnt=0;
    while(x)
    {
        x=x&(x-1);
        cnt++;
    }
    return cnt;
}

void solve()
{
    ll n,i,j,x;
    set<ll> s[65];
    cin >> n;
    while(n--)
    {
        cin >> x;
        s[nosb(x)].insert(x);
    }
    for(i=0;i<65;i++)
    {
        if(s[i].size())
        {
            set<ll> :: iterator it;
            for(it=s[i].begin();it!=s[i].end();it++) cout << *it << ' ';
            nl;
        }
    }
}

int main()
{
    opt;
    solve();
    return 0;
}

```

▲ 1 vote ● Reply ● Message ● Permalink

**Qi Chen** a year ago

?

```

#include <iostream>
using namespace std;
int counter(int a)
{
    int num = a;
    int counter = 0;
    while(num)
    {
        counter += num % 2;
        num = num / 2;
    }
    return counter;
}
void quick_sort(int a[], int l, int r)
{
    if(l<r)
    {
        int i=l, j=r;
        int x=a[l];
        while(i<j)
        {
            while(i<j && a[j]>=x)
                j--;
            if(i<j)
                a[i++] = a[j];
            while(i<j && a[i]<x)
                i++;
            if(i<j)
                a[j--] = a[i];
        }
        a[i] = x;
        quick_sort(a, l, i-1);
        quick_sort(a, i+1, r);
    }
}
int main()
{
    int N;
    cin >> N;
    int a[N];
    int num[9] = {0};
    int CC[9][1000];
    int count;
    for(int i=0; i<N; i++)
    {
        cin >> a[i];
        count = counter(a[i]);
        num[count-1] ++;
        CC[count-1][num[count-1]-1] = a[i];
    }
    for(int i=0; i<9; i++)
    {
        if(num[i] != 0)
            quick_sort(CC[i], 0, num[i]-1);
    }
    for(int i=0; i<9; i++)
    {
        if(num[i] !=0)

```

?

```
for(int j=0; j<num[i]; j++)
{
    cout << CC[i][j] <<" ";
}
cout << endl;
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink



RP a year ago

Do not post solutions in comments. Hints are fine. Doubts too. Plain solution just spoils the question.

▲ 0 votes ● Reply ● Message ● Permalink



nitish sharma a year ago

ok

▲ 0 votes ● Reply ● Message ● Permalink



AMAAN ANWAR a year ago

```
#include<bits/stdc++.h>
using namespace std;
```

```
typedef struct node
{
    int data;
    node *next;
```

```
}node;
node *x[10];
int unitp(int x)
{
    int rem,rev;
    while (x > 0)
    {
        rem = x % 10;
        rev = rev * 10 + rem;
        x /= 10;
    }
    return rev%10;
```

```
}
void linkk(node *x,int p ,int d)
{
    node *temp;
    temp->data=d;
    temp->next=NULL;
    x[p]->next=temp;
}
```

```
void insertionSort(node *arr, int n)
{
    int i, key, j;
    for (i = 1; i < n; i++)
    {
        key = arr[i];
        j = i-1;
```

```
while (j >= 0 && arr[j] > key)
```

?

```

{
    arr[j+1] = arr[j];
    j = j-1;
}
arr[j+1] = key;
}
}
void bucketSort(int arr[], int n)
{
    int up;
    for(int i=0;i<n;i++)
    {
        up=unitp(arr[i]);

        linkk(x[],up,arr[i]);
    }
    for(int j=0;j<n;j++)
    {
        int count=0;
        while (x[j]!=NULL)
        {
            x[j]=x[j]->next;
            count++;
        }
        if(count>0)
            insertionSort( x[j], count)
        }
        for(int j=0;j<n;j++)
        {
            while (x[j]!=NULL)
            {
                cout<<x[j]->data;
                x[j]=x[j]->next;
            }

        }

    }

    int main()
    {
        int n;
        cin>>n;
        int arr[n];
        for(int i=0;i<n;i++)
            cin>>arr[i];

        bucketSort(arr, n);

        cout << "Sorted array is \n";
        for (int i=0; i<n; i++)
            cout << arr[i] << " ";
        return 0;
    }

```

▲ 0 votes ● Reply ● Message ● Permalink



Rishabh Sairawat a year ago

Here is my Java Solution:

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

?


```

class TestClass {
private static void bucketSort(int[] arr) {
int n=10;
LinkedList<Integer>[] bucket = new LinkedList[n];
for (int i=0; i<n; i++)
bucket[i]=new LinkedList<Integer>();
for (int i=0; i<arr.length; i++)
bucket[setBits(arr[i])].add(arr[i]);
for (int i=0; i<n; i++){
if(bucket[i].size()>1)
Collections.sort(bucket[i]);
}
for (int i=0; i<n; i++){
if(bucket[i].size()>0){
for(Integer x:bucket[i]){
System.out.print(x+" ");
}
System.out.println();
}
}
}
private static int setBits(int num){
int count=0;
while(num!=0){
num &= (num-1);
count++;
}
return count;
}
public static void main(String args[] ) throws Exception {
Scanner s = new Scanner(System.in);
int N = s.nextInt();
int[] arr=new int[N];
for (int i = 0; i < N; i++) {
arr[i]=s.nextInt();
}
bucketSort(arr);
}
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Vanshika Shrivastava a year ago

here is my solution:

```

#include <iostream>
#include <vector>
#include <algorithm>
using namespace std;
typedef vector< vector<int> > vvi;
int count(int k)
{
int dem = 0;
for (int i = 0; i<10; i++)
{
if (k & 1 << i)
dem++;
}
return dem;
}
void bucketSort(int a[], int n, vvi b)
{

```

?

```

for (int i = 0; i<n; i++)
{
    b[count(a[i])].push_back(a[i]);
}
for (int i = 0; i<10; i++)
{
    sort(b[i].begin(), b[i].end());
}
for (int i = 0; i<10; i++)
{
    for (int j = 0; j<b[i].size(); j++)
    {
        cout << b[i][j];
        if (j < b[i].size() - 1)
            cout << " ";
    }
    if (i < n - 1)
        cout << endl;
}
}
int main()
{
    int n;
    int a[1010];
    vvi b;
    b.resize(11);
    cin >> n;
    for (int i = 0; i<n; i++)
        cin >> a[i];
    bucketSort(a, n, b);
    return 0;
}

```

▲ 0 votes ● Reply ● Message ● Permalink



Chaymae Ahmed 7 months ago

Numbers in each bucket must have more close values this how bucket sort works
How does the output of the first bucket have number 4 and the second bucket have number 3 in the "in the expected output sample"??

▲ 0 votes ● Reply ● Message ● Permalink



Tejas Duseja 3 months ago

Can anyone suggest substitute for ArrayList i used in my program

```

import java.util.*;
public class TestClass {
    public static int set_bits(int n){
        int set=0;
        while(n>0){
            if(n%2==1)
                set++;
            n=n/2;
        }
        return set;
    }
    public static int countbucket(int max){
        int bucket=0;
        while(max>0){
            bucket++;
            max=max/2;
        }
        return bucket;
    }
}

```

?

```
}
public static void main(String args[] ) throws Exception {
Scanner sc=new Scanner(System.in);
int s=sc.nextInt();
int ar[]=new int[s];
int max=0;
for(int i=0;i<s;i++){
ar[i]=sc.nextInt();
if(ar[i]>max)
max=ar[i];

}
int total_buckets=countbucket(max);
ArrayList<Integer> buckets[] =new ArrayList[total_buckets];
for(int i=0;i<total_buckets;i++){
buckets[i]=new ArrayList<Integer>();
}
for(int i=0;i<s;i++){
buckets[set_bits(ar[i])-1].add(ar[i]);
}
for(ArrayList a: buckets){
Collections.sort(a);
for(int i=0;i<a.size();i++)
System.out.print(a.get(i)+" ");
System.out.println();
}
}
}
```

▲ 0 votes ● Reply ● Message ● Permalink

[About Us](#)

[Innovation Management](#)

[Technical Recruitment](#)

[University Program](#)

[Developers Wiki](#)

[Blog](#)

[Press](#)

[Careers](#)

[Reach Us](#)



Site Language: [English](#) ▼ | [Terms and Conditions](#) | [Privacy](#) | © 2018 HackerEarth