All Tracks › Algorithms › Sorting › Bucket Sort

6 LIVE EVENTS

Algorithms

Solve any problem to achieve a rank

View Leaderboard

Topics: Bucket Sort

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 $O(NlogN)$. 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 distnict 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.