

**Write an  
Article**

**Login**

Quick Links for  
Sorting

Sorting  
Terminology

Stability in  
sorting  
algorithms

Time  
Complexities of  
all Sorting  
Algorithms

External Sorting

Sorting  
Algorithms

Bubble Sort,  
Recursive Bubble  
Sort

Insertion Sort,  
Recursive  
Insertion Sort

Selection Sort

Merge Sort,  
Iterative Merge  
Sort

3 - Way Merge  
Sort

Quick Sort,  
Iterative Quick  
Sort

3 - Way Quick  
Sort

Heap Sort

Counting Sort

Radix Sort



Bucket Sort
Shell Sort
Tim Sort
Bitonic Sort
Binary Insertion Sort
Comb Sort
Cycle Sort
Pancake Sorting
Pigeonhole Sort
Gnome Sort
Tree Sort
Explore More...
Library Implementations of Sorting Algorithms :
Comparator function of qsort() in C
std::sort() in C++ STL
C qsort() vs C++ sort()
Arrays.sort() in Java with examples
Collections.sort() in Java with Examples
Know Your Sorting Algorithm :
Sorting Weapons used by Programming Languages
Introsort- C++'s Sorting Weapon



Misc :
Hoare's vs Lomuto partition scheme in QuickSort
Serial Sort v/s Parallel Sort in Java
Lower bound for comparison based sorting algorithms
Which sorting algorithm makes minimum number of memory writes?
Where is Heap Sort used practically?
Find memory conflicts among multiple threads
Explore More...
Searching
Linear Search
Binary Search
Fibonacci Search
More...
Arrays
Segregate 0s and 1s in an array
Rearrange positive and negative numbers in $O(n)$ time and $O(1)$ extra space
Maximum Product Sub-array
More...
Strings



Return maximum occurring character in an input string
Check whether two strings are anagram of each other
Remove all duplicates from a given string
More...
Linked List
Detect loop in a linked list
Find the middle of a given linked list
Reverse a linked list
More...
Stack
Implement two stacks in an array
Infix to Postfix Conversion using Stack
The Stock Span Problem
More...
Queue
Implement Queue using Stacks
Implementation of Deque using circular array
How to efficiently implement k Queues in a single array?
More...



Binary Tree
Print level order traversal line by line   Set 1
Diagonal Traversal of Binary Tree
Boundary Traversal of binary tree
More...
Binary Search Tree
Lowest Common Ancestor in a Binary Search Tree
Check if each internal node of a BST has exactly one child
Merge Two Balanced Binary Search Trees
More...
Heap
Binomial Heap
Fibonacci Heap
More...
Graph
Detect Cycle in an Undirected Graph
Longest Path in a Directed Acyclic Graph
More...
Hashing
Count distinct elements in every window of size k



Largest subarray with equal number of 0s and 1s

More...

## Bucket Sort

Bucket sort is mainly useful when input is uniformly distributed over a range. For example, consider the following problem.

2.2

Sort a large set of floating point numbers which are in range from 0.0 to 1.0 and are uniformly distributed across the range. How do we sort the numbers efficiently?

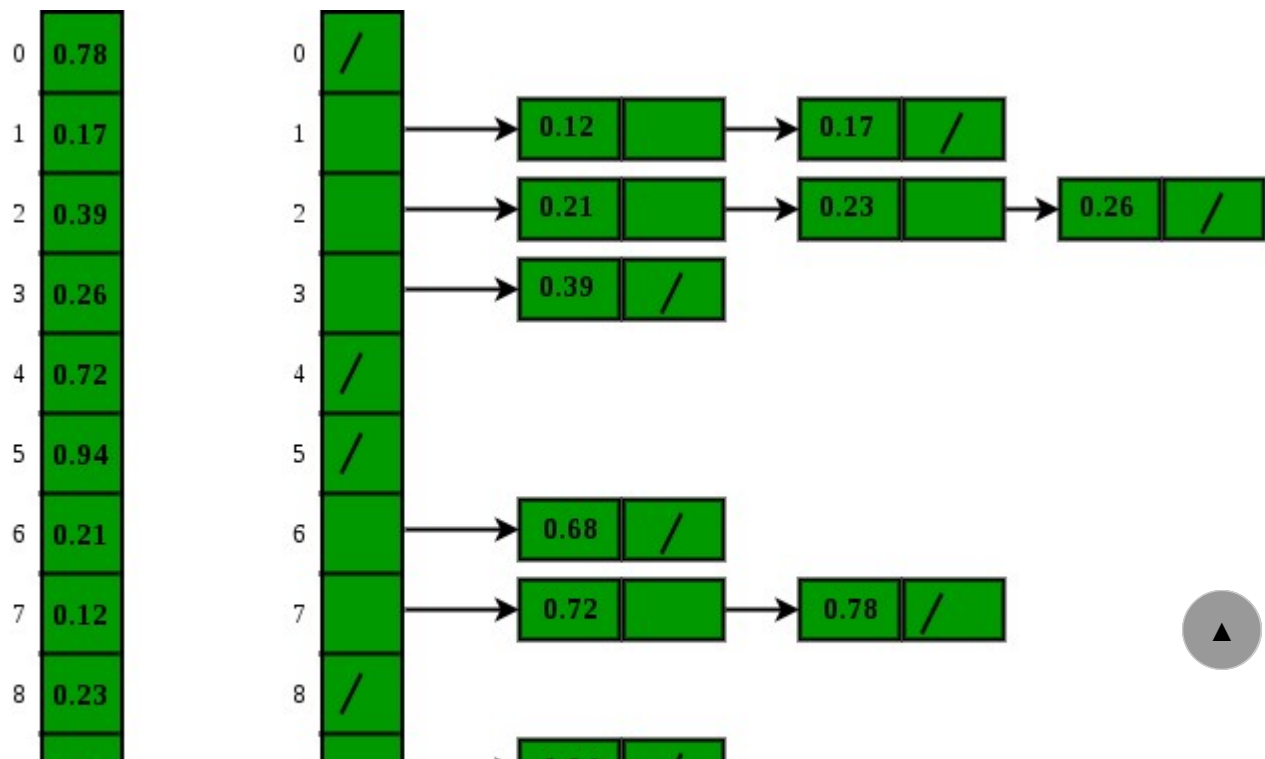
A simple way is to apply a comparison based sorting algorithm. The **lower bound for Comparison based sorting algorithm** (Merge Sort, Heap Sort, Quick-Sort .. etc) is  $\Omega(n \log n)$ , i.e., they cannot do better than  $n \log n$ .

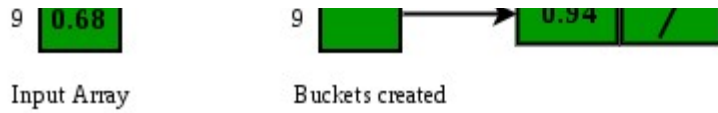
Can we sort the array in linear time? **Counting sort** can not be applied here as we use keys as index in counting sort. Here keys are floating point numbers.

The idea is to use bucket sort. Following is bucket algorithm.

**bucketSort(arr[], n)**

- 1) Create n empty buckets (Or lists).
- 2) Do following for every array element arr[i].  
.....a) Insert arr[i] into bucket[n\*array[i]]
- 3) Sort individual buckets using insertion sort.
- 4) Concatenate all sorted buckets.





Time Complexity: If we assume that insertion in a bucket takes  $O(1)$  time then steps 1 and 2 of the above algorithm clearly take  $O(n)$  time. The  $O(1)$  is easily possible if we use a linked list to represent a bucket (In the following code, C++ vector is used for simplicity). Step 4 also takes  $O(n)$  time as there will be  $n$  items in all buckets.

The main step to analyze is step 3. This step also takes  $O(n)$  time on average if all numbers are uniformly distributed (please refer [CLRS book](#) for more details)

Following is C++ implementation of the above algorithm.

```
// C++ program to sort an array using bucket sort
#include <iostream>
#include <algorithm>
#include <vector>
using namespace std;

// Function to sort arr[] of size n using bucket sort
void bucketSort(float arr[], int n)
{
    // 1) Create n empty buckets
    vector<float> b[n];

    // 2) Put array elements in different buckets
    for (int i=0; i<n; i++)
    {
        int bi = n*arr[i]; // Index in bucket
        b[bi].push_back(arr[i]);
    }

    // 3) Sort individual buckets
    for (int i=0; i<n; i++)
        sort(b[i].begin(), b[i].end());

    // 4) Concatenate all buckets into arr[]
    int index = 0;
    for (int i = 0; i < n; i++)
        for (int j = 0; j < b[i].size(); j++)
            arr[index++] = b[i][j];
}

/* Driver program to test above funtion */
int main()
{
    float arr[] = {0.897, 0.565, 0.656, 0.1234, 0.665, 0.3434};
    int n = sizeof(arr)/sizeof(arr[0]);
    bucketSort(arr, n);

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

Output:

Sorted array is

0.1234 0.3434 0.565 0.656 0.665 0.897

## Bucket Sort To Sort an Array with Negative Numbers

### References:

Introduction to Algorithms 3rd Edition by Clifford Stein, Thomas H. Cormen, Charles E. Leiserson, Ronald L. Rivest

[http://en.wikipedia.org/wiki/Bucket\\_sort](http://en.wikipedia.org/wiki/Bucket_sort)

## 连接超时

www.youtube.com 的服务器响应时间过长。

- 此站点暂时无法使用或者太过忙碌。请过几分钟后再试。
- 如果您无法载入任何网页，请检查您计算机的网络连接状态。
- 如果您的计算机或网络受到防火墙或者代理服务器的保护，请确认 Firefox 已被授权访问网络。

重试

Snapshots:





0.78 0.17 0.39 0.26 0.72 0.94 0.21 0.12 0.23 0.68

Total no. of elements are  $n = 10$ . So we create 10 buckets.

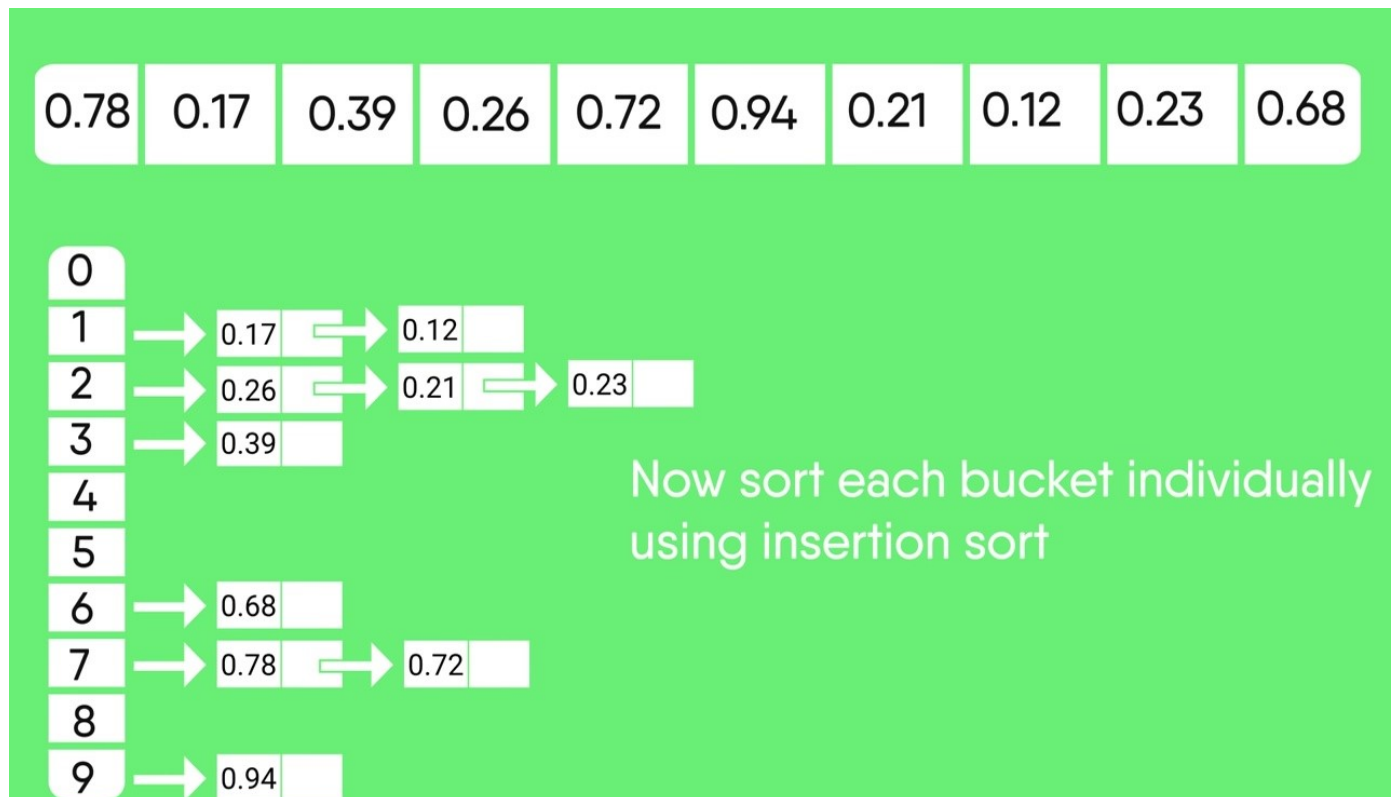
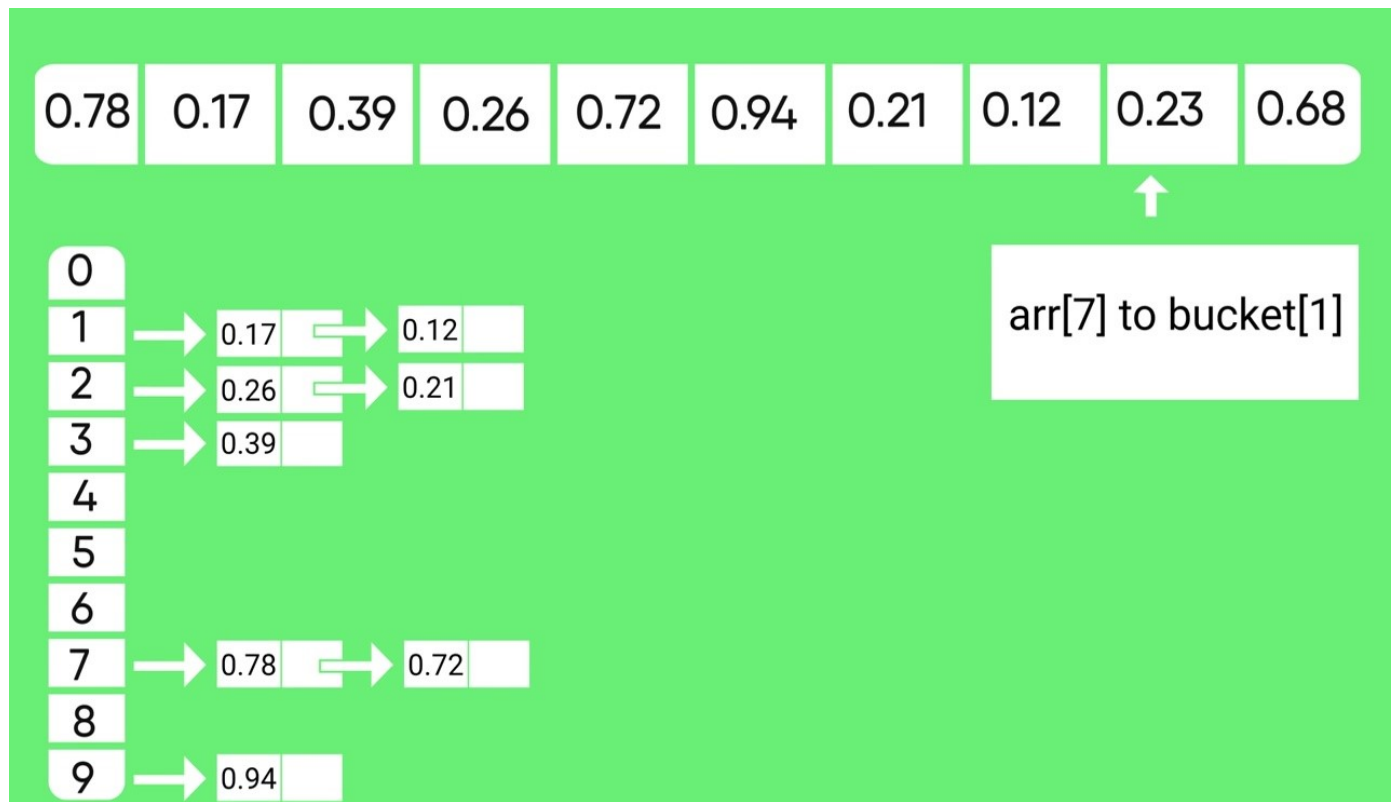
0.78 0.17 0.39 0.26 0.72 0.94 0.21 0.12 0.23 0.68

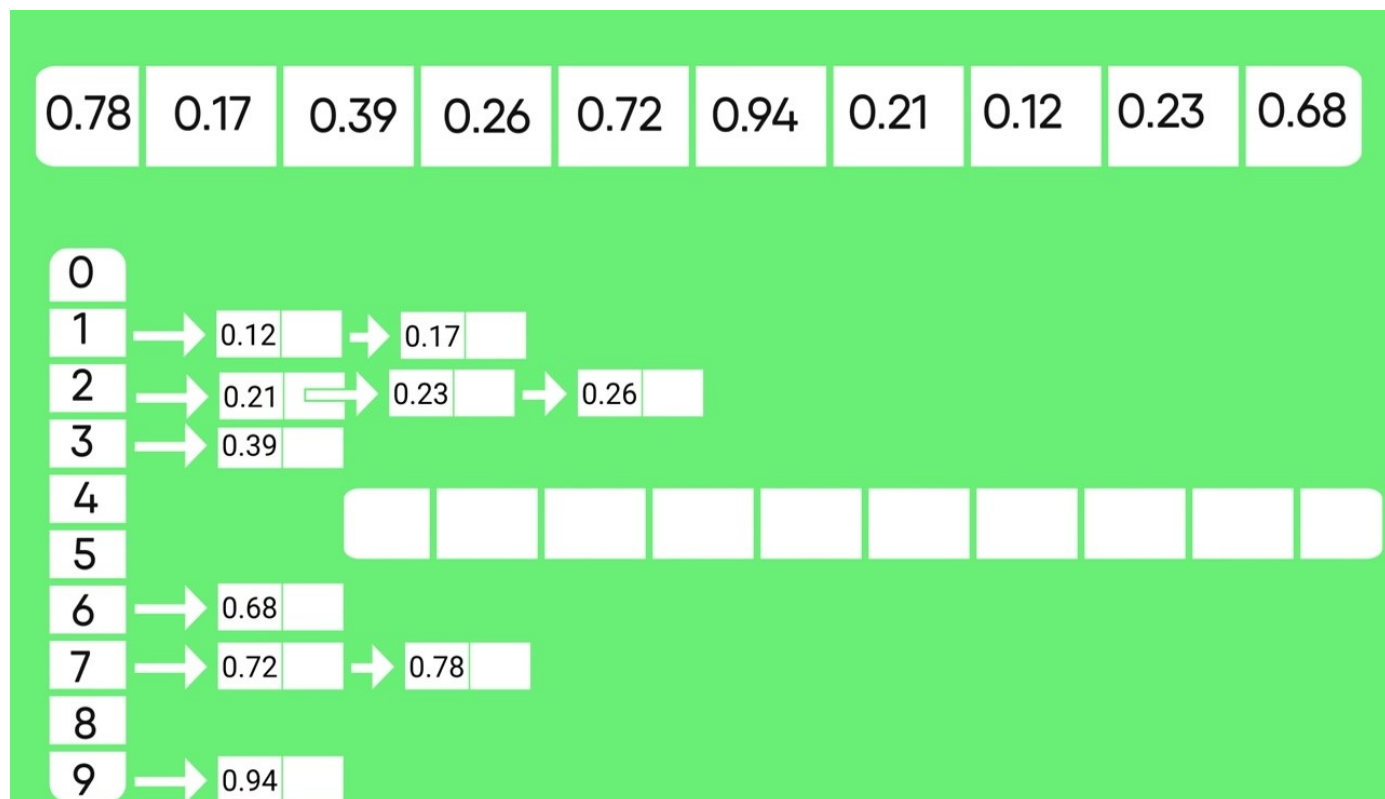
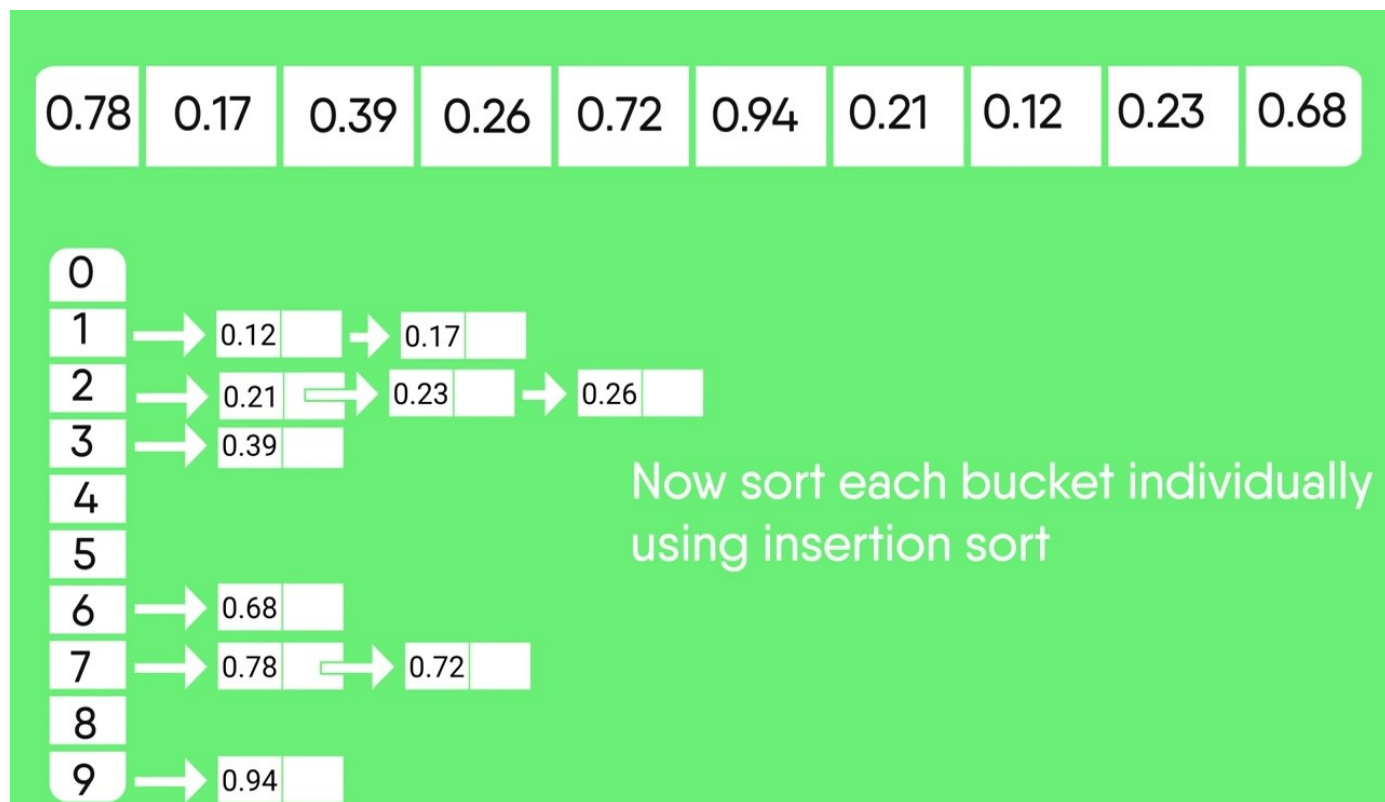


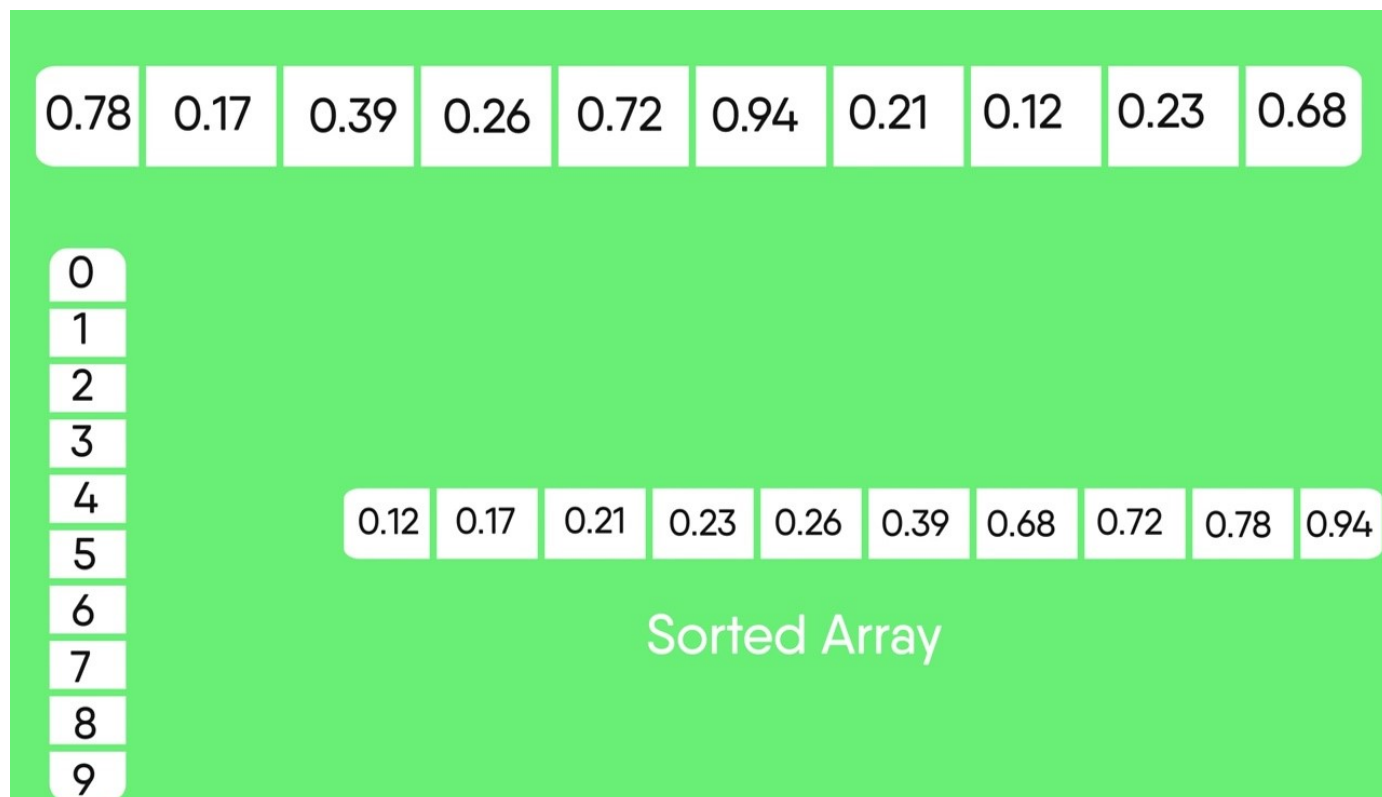
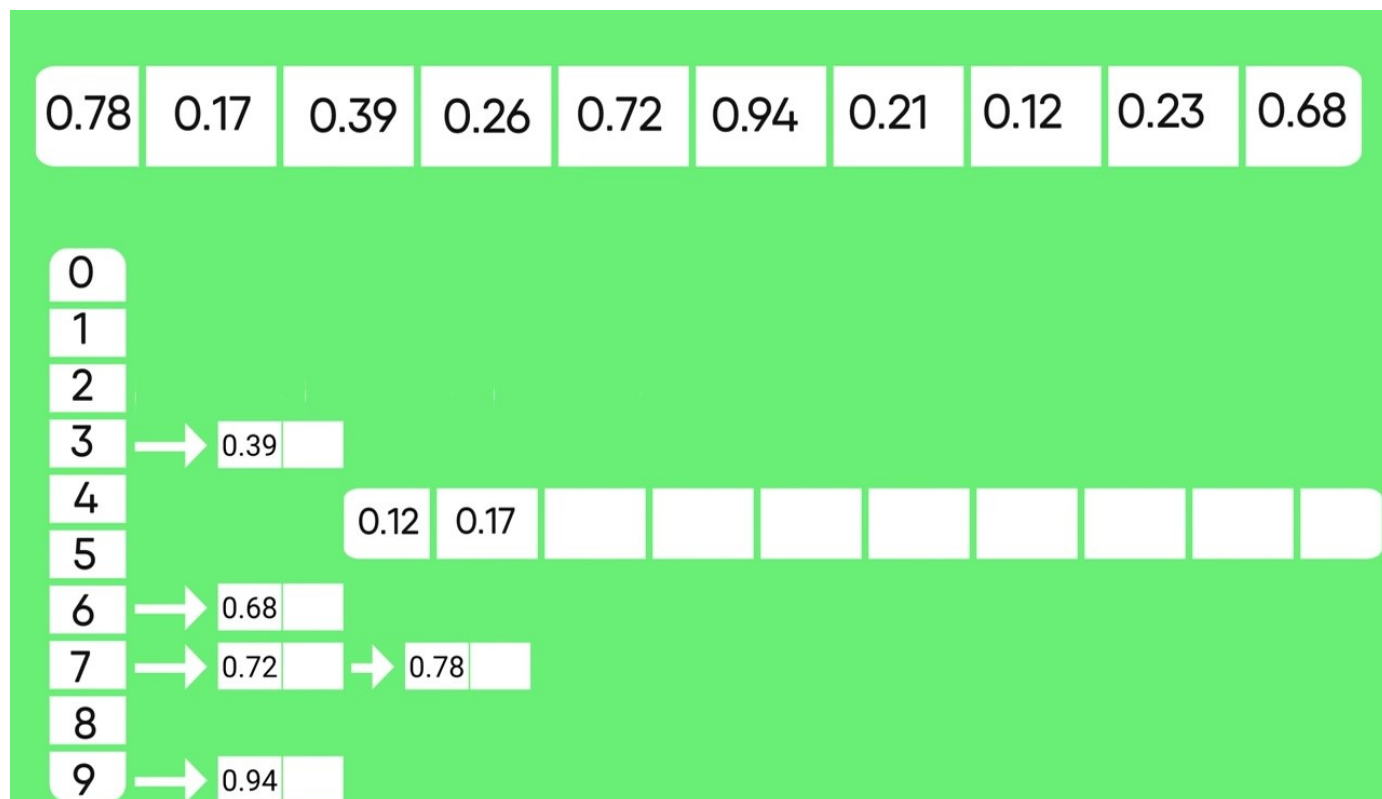
0	
1	→ 0.17
2	
3	→ 0.39
4	
5	
6	
7	→ 0.78
8	
9	

arr[2] to bucket[3]









## Quiz on Bucket Sort

Other Sorting Algorithms on GeeksforGeeks/GeeksQuiz:



- Selection Sort
- Bubble Sort
- Insertion Sort
- Merge Sort
- Heap Sort
- QuickSort
- Radix Sort
- Counting Sort
- ShellSort

Please write comments if you find anything incorrect, or you want to share more information about the topic discussed above



## GATE CS Corner   Company Wise Coding Practice

Sorting

[Login to Improve this Article](#)

Please write to us at [contribute@geeksforgeeks.org](mailto:contribute@geeksforgeeks.org) to report any issue with the above content.

### Recommended Posts:

[Radix Sort](#)

[ShellSort](#)

[Counting Sort](#)

[Comb Sort](#)

[Bucket Sort To Sort an Array with Negative Numbers](#)

[Sort the given string using character search](#)

[Pairs such that one is a power multiple of other](#)

[Loop Invariant Condition with Examples of Sorting Algorithms](#)

[Sort on the basis of number of factors using STL](#)

[Insertion sort to sort even and odd positioned elements in different orders](#)



([Login](#) to Rate)

2.2

Average Difficulty : 2.2/5.0  
Based on 47 vote(s)



Add to TODO List



Mark as DONE

Basic

Easy

Medium

Hard

Expert

Writing code in comment? Please use [ide.geeksforgeeks.org](https://ide.geeksforgeeks.org), generate link and share the link here.

Load Comments

Share this post!

A computer science portal for geeks

710-B, Advant Navis Business Park,  
Sector-142, Noida, Uttar Pradesh - 201305  
[feedback@geeksforgeeks.org](mailto:feedback@geeksforgeeks.org)

#### COMPANY

About Us  
Careers  
Privacy Policy  
Contact Us

#### LEARN

Algorithms  
Data Structures  
Languages  
CS Subjects  
Video Tutorials

#### PRACTICE

Company-wise  
Topic-wise  
Contests  
Subjective Questions

#### CONTRIBUTE

Write an Article  
GBlog  
Videos

@geeksforgeeks, Some rights reserved

