

Course Name: A Level (2nd Sem)**Topic:** Bucket Sort in C++**Subject:** Data Structure using C++**Date:** 21-04-2020**Example1 : Bucket Sort.**

In the Bucket Sorting technique, the data items are distributed of a set of buckets. Each bucket can hold similar type of data. After distributing, each bucket is sorted using another sorting algorithm. After that all elements are gathered into the main list to get the sorted form.

The complexity of Bucket Sort Technique

Time Complexity: $O(n + k)$ for best case and average case and $O(n^2)$ for worst case.

Space Complexity: $O(nk)$ for worst case

Input – A list of unsorted data: 0.25 0.36 0.58 0.41 0.29 0.22 0.45 0.79 0.01 0.69

Output – Array after Sorting: 0.01 0.22 0.25 0.29 0.36 0.41 0.45 0.58 0.69 0.79

Input: An array of data, and the total number in the array

Output: The sorted Array

Begin

```
for i := 0 to size-1 do
    insert array[i] into the bucket index (size * array[i])
done
for i := 0 to size-1 do
    sort bucket[i]
done
for i := 0 to size -1 do
    gather items of bucket[i] and put in array
done
End
```

```
#include<iostream>
#include<vector>
#include<algorithm>
using namespace std;
void display(float *array, int size)
{
    for(int i = 0; i<size; i++)
        cout << array[i] << " ";
    cout << endl;
}
void bucketSort(float *array, int size)
{
    vector<float> bucket[size];
    for(int i = 0; i<size; i++)
    {
        //put elements into different buckets
        bucket[int(size*array[i])].push_back(array[i]);
    }
    for(int i = 0; i<size; i++)
    {
        sort(bucket[i].begin(), bucket[i].end());    //sort individual vectors
    }
}
```

```

    }
    int index = 0;
    for(int i = 0; i<size; i++)
    {
        while(!bucket[i].empty())
        {
            array[index++] = *(bucket[i].begin());
            bucket[i].erase(bucket[i].begin());
        }
    }
}

int main()
{
    int n;
    cout << "Enter the number of elements: ";
    cin >> n;
    float arr[n];    //create an array with given number of elements
    cout << "Enter elements:" << endl;
    for(int i = 0; i<n; i++)
    {
        cin >> arr[i];
    }
    cout << "Array before Sorting: ";
    display(arr, n);
    bucketSort(arr, n);
    cout << "Array after Sorting: ";
    display(arr, n);
}

```

Output

Enter the number of elements: 10

Enter elements: 0.25 0.36 0.58 0.41 0.29 0.22 0.45 0.79 0.01 0.69

Array before Sorting: 0.25 0.36 0.58 0.41 0.29 0.22 0.45 0.79 0.01
0.69

Array after Sorting: 0.01 0.22 0.25 0.29 0.36 0.41 0.45 0.58 0.69
0.79