Convert an array to reduced form

Given an array with n distinct elements, convert the given array to a form where all elements are in range from 0 to n-1. The order of elements is same, i.e., 0 is placed in place of smallest element, 1 is placed for second smallest element, … n-1 is placed for largest element.

```
Input:  arr[] = {10, 40, 20}
Output: arr[] = {0, 2, 1}

Input:  arr[] = {5, 10, 40, 30, 20}
Output: arr[] = {0, 1, 4, 3, 2}
```

Expected time complexity is O(n Log n).

We strongly recommend that you click here and practice it, before moving on to the solution.

Method 1 (Simple)

A Simple Solution is to first find minimum element replace it with 0, consider remaining array and find minimum in the remaining array and replace it with 1 and so on. Time complexity of this solution is O(n^2)

Method 2 (Efficient)

The idea is to use hashing and sorting. Below are steps.
1) Create a temp array and copy contents of given array to temp[]. This takes O(n) time.
2) Sort temp[] in ascending order. This takes O(n Log n) time.
3) Create an empty hash table. This takes O(1) time.
4) Traverse temp[] form left to right and store mapping of numbers and their values (in converted array) in hash table. This takes O(n) time on average.
5) Traverse given array and change elements to their positions using hash table. This takes O(n) time on average.

Overall time complexity of this solution is O(n Log n).

Below is C++ implementation of above idea.

```
// C++ program to convert an array in reduced
// form
#include <bits/stdc++.h>
using namespace std;

void convert(int arr[], int n)
{
    // Create a temp array and copy contents
    // of arr[] to temp
    int temp[n];
    memcpy(temp, arr, n*sizeof(int));

    // Sort temp array
    sort(temp, temp + n);

    // Create a hash table. Refer
    // http://tinyurl.com/zp5wgef
    unordered_map<int, int> umap;

    // One by one insert elements of sorted
    // temp[] and assign them values from 0
    // to n-1
    int val = 0;
    for (int i = 0; i < n; i++)
        umap[temp[i]] = val++;

    // Convert array by taking positions from
    // umap
    for (int i = 0; i < n; i++)
        arr[i] = umap[arr[i]];
}

void printArr(int arr[], int n)
{
    for (int i=0; i<n; i++)
        cout << arr[i] << " ";
}

// Driver program to test above method
int main()
{
    int arr[] = {10, 20, 15, 12, 11, 50};
    int n = sizeof(arr)/sizeof(arr[0]);

    cout << "Given Array is \n";
    printArr(arr, n);

    convert(arr , n);

    cout << "\n\nConverted Array is \n";
    printArr(arr, n);

    return 0;
}
```

Output :

```
Given Array is
10 20 15 12 11 50

Converted Array is
0 4 3 2 1 5
```