# Department of Computing

# School of Electrical Engineering and Computer Science

**CS-250: Data Structure and Algorithms**

**Class: BSCS 10C**

**Lab 10:  Counting Sort Problems**

**Date: 26th November, 2021**

**Time: 9:00 am – 11:50 am**

# Instructor: Prof. Dr. Faisal Shafait

# Lab Engineer: Mr. Aftab Farooq

**Lab 10: Counting Sort Problems**

**Introduction**

For certain types of input, it is more efficient to use a non-comparison sorting algorithm. This will make it possible to sort lists even in linear time. These challenges will cover Counting Sort, a fast way to sort lists where the elements have a small number of possible values, such as integers within a certain range

**Objectives**

Objective of this lab is to implement Counting Sort Algorithm to solve different problems.

**Tools/Software Requirement**

Visual Studio C++

**Lab Tasks**

**Task 1:**

Given a list of integers, can you count and output the number of times each value appears?

Hint: There is no need to sort the data, you just need to count it.

Input Format

There will be two lines of input:

n- the size of the list

ar - n space-separated numbers that make up the list

Output Format

Output the number of times every number from to (inclusive) appears on the list.

Sample Input

100

63 25 73 1 98 73 56 84 86 57 16 83 8 25 81 56 9 53 98 67 99 12 83 89 80 91 39 86 76 85 74 39 25 90 59 10 94 32 44 3 89 30 27 79 46 96 27 32 18 21 92 69 81 40 40 34 68 78 24 87 42 69 23 41 78 22 6 90 99 89 50 30 20 1 43 3 70 95 33 46 44 9 69 48 33 60 65 16 82 67 61 32 21 79 75 75 13 87 70 33

Sample Output

0 2 0 2 0 0 1 0 1 2 1 0 1 1 0 0 2 0 1 0 1 2 1 1 1 3 0 2 0 0 2 0 3 3 1 0 0 0 0 2 2 1 1 1 2 0 2 0 1 0 1 0 0 1 0 0 2 1 0 1 1 1 0 1 0 1 0 2 1 3 2 0 0 2 1 2 1 0 2 2 1 2 1 2 1 1 2 2 0 3 2 1 1 0 1 1 1 0 2 2

Explanation

The output states that 0 appears 0 times, 1 appears 2 times, 2 appears 0 times, and so on in the given input array.

Link to problem : <https://www.hackerrank.com/challenges/countingsort1/problem>

**Code:**

|  |
| --- |
| #include<iostream>  using namespace std;  int main() {  int arr[100];  int duplicates[101];  int size;  cout << "Enter the size of the Array :" << endl;  cin >> size;  cout << "Enter the elements of the Array : " << endl;  for (int i = 0; i < size; i++) {  cin >> arr[i];  }  cout << "The Entered Array is : " << endl;  for (int i = 0; i < size; i++) {  cout << arr[i] << " ";  }  for (int i = 0; i < size; i++) {  duplicates[i] = 0;  }//using count sort to calculate the duplicates  for (int i = 0; i < size; i++) {  duplicates[arr[i]]++;  }  cout << endl;  cout << "The duplicates in the Array are: " << endl;  for (int i = 0; i < size; i++) {  cout << duplicates[i] << " ";  }  } |

**Output:**

**Graphical user interface, text

Description automatically generated**

**Task 2:**

Given an unsorted list of integers, output the integers in order.

Hint: You can use your previous code that counted the items to print out the actual values in order.

**Sample Input**

100

63 25 73 1 98 73 56 84 86 57 16 83 8 25 81 56 9 53 98 67 99 12 83 89 80 91 39 86 76 85 74 39 25 90 59 10 94 32 44 3 89 30 27 79 46 96 27 32 18 21 92 69 81 40 40 34 68 78 24 87 42 69 23 41 78 22 6 90 99 89 50 30 20 1 43 3 70 95 33 46 44 9 69 48 33 60 65 16 82 67 61 32 21 79 75 75 13 87 70 33

**Sample Output**

1 1 3 3 6 8 9 9 10 12 13 16 16 18 20 21 21 22 23 24 25 25 25 27 27 30 30 32 32 32 33 33 33 34 39 39 40 40 41 42 43 44 44 46 46 48 50 53 56 56 57 59 60 61 63 65 67 67 68 69 69 69 70 70 73 73 74 75 75 76 78 78 79 79 80 81 81 82 83 83 84 85 86 86 87 87 89 89 89 90 90 91 92 94 95 96 98 98 99 99

Explanation In the output you can see the numbers sorted in ascending order. You can also see that numbers appearing multiple times are printed accordingly.

Link: <https://www.hackerrank.com/challenges/countingsort2/problem>

**Code:**

|  |
| --- |
| #include<iostream>  #include<algorithm>  using namespace std;  void display(int\* arr, int size) {  //Function to disolay the array  for (int i = 1; i <= size; i++)  cout << arr[i] << " ";  cout << endl;  }  int getMax(int arr[], int size) {  int max = arr[1];  for (int i = 2; i <= size; i++) {  if (arr[i] > max)  max = arr[i];  }  return max; //the max element from the array  }  void countSort(int\* arr, int size) {  //Code from geeks for geeks as it is already inveted  //Programming Rule don't invent a new wheel just modified it according  //to my needs  int result[ 101];  int max = getMax(arr, size);  int count[ 101]; /\*  As we cannot intialize it with max+1 that's why  count and result are initialized with 101(dummy  \*/  for (int i = 0; i <= max; i++)  {  count[i] = 0;  }  for (int i = 1; i <= size; i++) {  count[arr[i]]++;//Calculating the duplicates  }    for (int i = 1; i <= max; i++) {  count[i] += count[i - 1];  }  //find cumulative frequency  for (int i = size; i >= 1; i--) {  result[count[arr[i]]] = arr[i];  count[arr[i]] -= 1; //decrease count for same numbers  }  for (int i = 1; i <= size; i++) {  arr[i] = result[i]; //store output array to main array  }  }  int main() {  int size;  cout << "Enter the number of elements: ";  cin >> size;  int arr[100]; //create an array with given number of elements  cout << "Enter elements:" << endl;  for (int i = 1; i <= size; i++) {  cin >> arr[i];  }  cout << "Array before Sorting: ";  display(arr, size);  countSort(arr, size);  cout << "Array after Sorting: ";  display(arr, size);  } |

**Output:**

**Text

Description automatically generated**

**Task 3:**

You will be given a list that contains both integers and strings. Can you print the strings in order of their accompanying integers? If the integers for two strings are equal, ensure that they are print in the order they appeared in the original list.

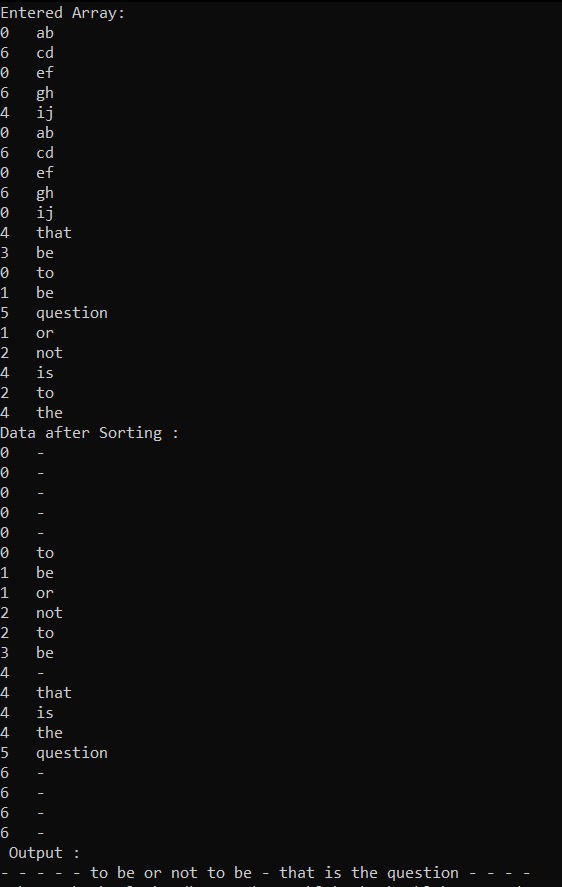
The Twist - Your clients just called with an update. They don't want you to print the first half of the original array. Instead, they want you to print a dash for any element from the first half.

Link: <https://www.hackerrank.com/challenges/countingsort4/problem>

**Code:**

|  |
| --- |
| #include<iostream>  #include<algorithm>  #include<string>  #include<vector>  using namespace std;  void display(int\* arr, int size) {  //Function to disolay the array  for (int i = 1; i <= size; i++)  cout << arr[i] << " ";  cout << endl;  }  vector<vector<string>> countSort(vector<vector<string>> arr) {  int size = arr.size();  vector<int> count(size);  for (int i = 0; i < size; i++) { //measuring frequency of each array  count[stoi(arr[i][0])]++;  }  for (int i = 1; i < count.size(); i++) {  count[i] += count[i - 1];  }  vector<vector<string>> sortedArray(size, vector<string>(arr[0].size()));  for (int i = size - 1; i >= 0; i--) {  sortedArray[count[stoi(arr[i][0])] - 1][0] = arr[i][0];  sortedArray[count[stoi(arr[i][0])] - 1][1] = arr[i][1];  count[stoi(arr[i][0])]--;  }  return sortedArray;  }  void print(vector<vector<string>> arr) {  for (int i = 0; i < arr.size(); i++) {  cout << arr[i][1] << " ";  }  }  void printTogether(vector<vector<string>> arr) {  for (int i = 0; i < arr.size(); i++) {  cout << arr[i][0] << " " << arr[i][1] << endl;  }  }  int main() {  vector<vector<string>> arr = {  {"0", "ab"},  {"6","cd"},  {"0", "ef"},  {"6", "gh"},  {"4", "ij"},  {"0", "ab"},  {"6", "cd"},  {"0", "ef"},  {"6", "gh"},  {"0", "ij"},  {"4", "that"},  {"3", "be"},  {"0", "to"},  {"1", "be"},  {"5", "question"},  {"1", "or" },  {"2", "not"},  {"4", "is"},  {"2", "to"},  {"4", "the"}  };  cout << "Entered Array: " << endl;  printTogether(arr);  for (int i = 0; i < arr.size() / 2; i++) {  //For printing half of the Array  arr[i][1] = "-";  }  vector<vector<string>> sortedArray = countSort(arr);  //Calling the funnction to have the sortedArray  cout << "Data after Sorting : " << endl;  printTogether(sortedArray);  cout << " Output : " << endl;  print(sortedArray);    } |

**Output:**



**Important Note:** Practice your knowledge of OOP with C++ when creating a solution.

**Lab Grading:**

|  |  |
| --- | --- |
| **Task** | **Marks** |
| Lab Viva/Quiz | 5 |
| Comments/ Indentation | 2 |
| Solution Document | 2 |
| Output Screen Shots | 1 |
| -- | -- |
| Total | 10 |

**Deliverables**

This lab grading policy is as follows: The lab is graded between 0 to 10 marks. The submitted solution can get a maximum of 5 marks. At the end of each lab or in the next lab, there will be a viva related to the tasks. The viva has a weightage of 5 marks. Insert the solution/answer in this document. You must show the implementation of the tasks in the designing tool, along with your complete Word document to get your work graded. You must also submit this Word document on the LMS. In case of any problems discuss it by emailing it to [aftab.farooq@seecs.edu.pk](mailto:aftab.farooq@seecs.edu.pk).

**Note:** Students are required to upload the lab on LMS before deadline.

Use proper indentation and comments. Lack of comments and indentation will result in deduction of marks.