Count Distinct (Unique) elements in an array - GeeksforGeeks

Source: https://www.geeksforgeeks.org/count-distinct-elements-in-an-array/

Courses Tutorials Practice Jobs DSA Tutorial Interview Questions Quizzes Must Do Advanced DSA System Design Aptitude Puzzles Interview Corner DSA Python Technical Scripter 2026 Explore DSA Fundamentals Logic Building Problems Analysis of Algorithms Data Structures Array Data Structure String in Data Structure Hashing in Data Structure Linked List Data Structure Stack Data Structure Queue Data Structure Tree Data Structure Graph Data Structure Trie Data Structure Algorithms Searching Algorithms Sorting Algorithms Introduction to Recursion Greedy Algorithms Tutorial Graph Algorithms Dynamic Programming or DP Bitwise Algorithms Advanced Segment Tree Binary Indexed Tree or Fenwick Tree Square Root (Sqrt) Decomposition Algorithm Binary Lifting Geometry Interview Preparation Interview Corner GfG160 Practice Problem GeeksforGeeks Practice - Leading Online Coding Platform Problem of The Day - Develop the Habit of Coding DSA Course 90% Refund Count Distinct (Unique) elements in an array Last Updated : 11 Jul, 2025 Given an array arr[] of length N , The task is to count all distinct elements in arr[] . Examples: Input: arr[] = {10, 20, 20, 10, 30, 10} Output: 3 Explanation: There are three distinct elements 10, 20, and 30. Input: arr[] = {10, 20, 20, 10, 20} Output: 2 Try it on GfG Practice Naïve Approach: Create a count variable and run two loops, one with counter i from 0 to N-1 to traverse arr[] and second with counter j from 0 to i-1 to check if i th element has appeared before. If yes , increment the count . Below is the Implementation of the above approach. C++ // C++ program to count distinct elements // in a given array #include <iostream> using namespace std ; int countDistinct (int arr [], int n) { int res = 1 ; // Pick all elements one by one for (int i = 1 ; i < n ; i ++) { int j = 0 ; for (j = 0 ; j < i ; j ++) if (arr [i] == arr [j]) break ; // If not printed earlier, then print it if (i == j) res ++ ; } return res ; } // Driver program to test above function int main () { int arr [] = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; int n = sizeof (arr) / sizeof (arr [0]); cout << countDistinct (arr , n); return 0 ; } Java // Java program to count distinct // elements in a given array import java.io.* ; class GFG { static int countDistinct (int arr [] , int n) { int res = 1 ; // Pick all elements one by one for (int i = 1 ; i < n ; i ++) { int j = 0 ; for (j = 0 ; j < i ; j ++) if (arr [i] == arr [j]) break ; // If not printed earlier, // then print it if (i == j) res ++ ; } return res ; } // Driver code public static void main (String [] args) { int arr [] = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; int n = arr . length ; System . out . println (countDistinct (arr , n)); } } // This code is contributed by Code_Mech. Python # Python3 program to count distinct # elements in a given array def countDistinct (arr , n): res = 1 # Pick all elements one by one for i in range (1 , n): j = 0 for j in range (i): if (arr [i] == arr [j]): break # If not printed earlier, then print it if (i == j + 1): res += 1 return res # Driver Code arr = [12 , 10 , 9 , 45 , 2 , 10 , 10 , 45] n = len (arr) print (countDistinct (arr , n)) # This code is contributed by Mohit Kumar C# // C# program to count distinct // elements in a given array using System ; class GFG { static int countDistinct (int [] arr , int n) { int res = 1 ; // Pick all elements one by one for (int i = 1 ; i < n ; i ++) { int j = 0 ; for (j = 0 ; j < i ; j ++) if (arr [i] == arr [j]) break ; // If not printed earlier, // then print it if (i == j) res ++ ; } return res ; } // Driver code public static void Main () { int [] arr = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; int n = arr . Length ; Console . WriteLine (countDistinct (arr , n)); } } // This code is contributed // by Akanksha Rai JavaScript < script > // JavaScript program to count distinct elements // in a given array function countDistinct (arr , n) { let res = 1 ; // Pick all elements one by one for (let i = 1 ; i < n ; i ++) { let j = 0 ; for (j = 0 ; j < i ; j ++) if (arr [i] === arr [j]) break ; // If not printed earlier, then print it if (i === j) res ++ ; } return res ; } // Driver program to test above function let arr = [12 , 10 , 9 , 45 , 2 , 10 , 10 , 45]; let n = arr . length ; document . write (countDistinct (arr , n)); // This code is contributed by Surbhi Tyagi < /script> PHP <?php // PHP program to count distinct elements // in a given array function countDistinct (& $arr , $n) { $res = 1 ; // Pick all elements one by one for ($i = 1 ; $i < $n ; $i ++) { for ($j = 0 ; $j < $i ; $j ++) if ($arr [$i] == $arr [$j]) break ; // If not printed earlier, // then print it if ($i == $j) $res ++ ; } return $res ; } // Driver Code $arr = array (12 , 10 , 9 , 45 , 2 , 10 , 10 , 45); $n = count ($arr); echo countDistinct ($arr , $n); // This code is contributed by // Rajput-Ji ?> Output 5 Time Complexity: O(n 2) Auxiliary Space: O(1) Using sorting : Below is the idea to solve the problem: Sort the array so that all occurrences of every element become consecutive. Once the occurrences become consecutive, then traverse the sorted array and count distinct elements by comparing the consecutive elements. Follow the steps below to

Implement the idea: Initialize a res variable with 0 and sort arr[] . Run a for loop from 0 to N-1 . While i is less than N-1 and arr[i] is equal to arr[i+1] increment i. increment res by one. Return res. Below is the implementation of above approach that is as follows: C++ // C++ program to count all distinct elements // in a given array #include <algorithm> #include <iostream> using namespace std ; int countDistinct (int arr [], int n) { // First sort the array so that all // occurrences become consecutive sort (arr , arr + n); // Traverse the sorted array int res = 0 ; for (int i = 0 ; i < n ; i ++) { // Move the index ahead while // there are duplicates while (i < n - 1 && arr [i] == arr [i + 1]) i ++ ; res ++ ; } return res ; } // Driver program to test above function int main () { int arr [] = { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; int n = sizeof (arr) / sizeof (arr [0]); cout << countDistinct (arr , n); return 0 ; } Java // Java program to count all distinct elements // in a given array import java.util.Arrays ; class GFG { static int countDistinct (int arr [] , int n) { // First sort the array so that all // occurrences become consecutive Arrays . sort (arr); // Traverse the sorted array int res = 0 ; for (int i = 0 ; i < n ; i ++) { // Move the index ahead while // there are duplicates while (i < n - 1 && arr [i] == arr [i + 1]) { i ++ ; } res ++ ; } return res ; } // Driver code public static void main (String [] args) { int arr [] = { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; int n = arr . length ; System . out . println (countDistinct (arr , n)); } } // This code is contributed by 29AjayKumar Python # Python3 program to count all distinct # elements in a given array def countDistinct (arr , n): # First sort the array so that all # occurrences become consecutive arr . sort () # Traverse the sorted array res = 0 i = 0 while (i < n): # Move the index ahead while # there are duplicates while (i < n - 1 and arr [i] == arr [i + 1]): i += 1 res += 1 i += 1 return res # Driver Code arr = [6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10] n = len (arr) print (countDistinct (arr , n)) # This code is contributed by mits C# // C# program to count all distinct elements // in a given array using System ; class GFG { static int countDistinct (int [] arr , int n) { // First sort the array so that all // occurrences become consecutive Array . Sort (arr); // Traverse the sorted array int res = 0 ; for (int i = 0 ; i < n ; i ++) { // Move the index ahead while // there are duplicates while (i < n - 1 && arr [i] == arr [i + 1]) { i ++ ; } res ++ ; } return res ; } // Driver code public static void Main () { int [] arr = { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; int n = arr . Length ; Console . WriteLine (countDistinct (arr , n)); } } // This code is contributed by Code_Mech. JavaScript < script > // JavaScript program to count all distinct elements // in a given array function countDistinct (arr , n) { // First sort the array so that all // occurrences become consecutive arr . sort (function (a , b){ return a - b ;}); // Traverse the sorted array let res = 0 ; for (let i = 0 ; i < n ; i ++) { // Move the index ahead while // there are duplicates while (i < n - 1 && arr [i] == arr [i + 1]) { i ++ ; } res ++ ; } return res ; } // Driver code let arr = [6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10]; let n = arr . length ; document . write (countDistinct (arr , n)); // This code is contributed by unknown2108 < /script> PHP <?php // PHP program to count all distinct // elements in a given array function countDistinct ($arr , $n) { // First sort the array so that all // occurrences become consecutive sort ($arr , 0); // Traverse the sorted array $res = 0 ; for ($i = 0 ; $i < $n ; $i ++) { // Move the index ahead while // there are duplicates while ($i < $n - 1 && $arr [$i] == $arr [$i + 1]) $i ++ ; $res ++ ; } return $res ; } // Driver Code $arr = array (6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10); $n = sizeof ($arr); echo countDistinct ($arr , $n); // This code is contributed by Akanksha Rai ?> Output 6 Time Complexity: O(n logn) Auxiliary Space: O(1) Using Hashing : The idea is to traverse the given array from left to right and keep track of visited elements in a hash set , as a set consists of only unique elements. Follow the steps below to implement the idea: Create an unordered_set s and a variable res initialized with 0. Run a for loop from 0 to N-1 If the current element is not present in s, insert it in s increment res by 1. Return res . Below is the implementation of the above approach. C++ /* CPP program to print all distinct elements of a given array */ #include <bits/stdc++.h> using namespace std ; // This function prints all distinct elements int countDistinct (int arr [], int n) { // Creates an empty hashset unordered_set < int > s ; // Traverse the input array int res = 0 ; for (int i = 0 ; i < n ; i ++) { // If not present, then put it in // hashtable and increment result if (s . find (arr [i]) == s . end ()) { s . insert (arr [i]); res ++ ; } } return res ; } // Driver Code int main () { int arr [] = { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; int n = sizeof (arr) / sizeof (arr [0]); cout << countDistinct (arr , n); return 0 ; } Java // Java Program to count // Unique elements in Array import java.util.* ; class GFG { // This method returns count // of Unique elements public static int countDistinct (int arr [] , int n) { HashSet < Integer > hs = new HashSet < Integer > (); for (int i = 0 ; i < n ; i ++) { // add all the elements to the HashSet hs . add (arr [i]); } // return the size of hashset as // it consists of all Unique elements return hs . size (); } // Driver code public static void main (String [] args) { int arr [] = new int [] { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; System . out . println (countDistinct (arr , arr . length)); } } // This code is contributed by Adarsh_Verma Python ''' Python3 program to print all distinct elements of a given array ''' # This function prints all distinct elements def countDistinct (arr , n): # Creates an empty hashset s = set () # Traverse the input array res = 0 for i in range (n): # If not present, then put it in # hashtable and increment result if (arr [i

] not in s): s . add (arr [i]) res += 1 return res # Driver code arr = [6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10] n = len (arr) print (countDistinct (arr , n)) # This code is contributed by SHUBHAMSINGH10 C# // C# Program to count // Unique elements in Array using System ; using System.Collections.Generic ; class GFG { // This method returns count // of Unique elements public static int countDistinct (int [] arr , int n) { HashSet < int > hs = new HashSet < int > (); for (int i = 0 ; i < n ; i ++) { // add all the elements to the HashSet hs . Add (arr [i]); } // return the size of hashset as // it consists of all Unique elements return hs . Count ; } // Driver code public static void Main () { int [] arr = new int [] { 6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10 }; Console . WriteLine (countDistinct (arr , arr . Length)); } } /* This code contributed by PrinciRaj1992 */ JavaScript < script > // Javascript Program to count // Unique elements in Array // This method returns count // of Unique elements function countDistinct (arr , n) { let hs = new Set (); for (let i = 0 ; i < n ; i ++) { // add all the elements to the HashSet hs . add (arr [i]); } // return the size of hashset as // it consists of all Unique elements return hs . size ; } // Driver code let arr = [6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10]; document . write (countDistinct (arr , arr . length)); // This code is contributed by patel2127 < /script> PHP <?php // PHP program to print all distinct elements // of a given array // This function prints all distinct elements function countDistinct ($arr , $n) { // Creates an empty hashset $s = array (); // Traverse the input array $res = 0 ; for ($i = 0 ; $i < $n ; $i ++) { // If not present, then put it in // hashtable and increment result array_push ($s , $arr [$i]); } $s = array_unique ($s); return count ($s); } // Driver Code $arr = array (6 , 10 , 5 , 4 , 9 , 120 , 4 , 6 , 10); $n = count ($arr); echo countDistinct ($arr , $n); // This code is contributed by mits ?> Output 6 Time complexity: O(n) Auxiliary Space: O(n), since n extra space has been taken. Using Set STL : Iterate over all the elements of the array insert them in an unordered set. As the set only contains distinct elements, so the size of set will be the answer. Follow the below steps to Implement the idea: Insert all the elements into the set S one by one. Store the total size s of the set using set::size() . The total size s is the number of distinct elements present in the array. Below is the Implementation of above approach. C++ #include <bits/stdc++.h> using namespace std ; // function that accepts the array and it's size and returns // the number of distince elements int distinct (int * arr , int len) { // declaring a set container using STL set < int > S ; for (int i = 0 ; i < len ; i ++) { // inserting all elements of the // array into set S . insert (arr [i]); } // calculating the size of the set int ans = S . size (); return ans ; } int main () { int arr [] = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; // calculating the size of the array int l = sizeof (arr) / sizeof (int); // calling the function on array int dis_elements = distinct (arr , l); cout << dis_elements << endl ; return 0 ; } Java import java.util.* ; class GFG { // function that accepts // the array and it's size and // returns the number of distince elements static int distinct (int [] arr , int len) { // declaring a set container // using STL HashSet < Integer > S = new HashSet <> (); for (int i = 0 ; i < len ; i ++) { // inserting all elements of the // array into set S . add (arr [i]); } // calculating the size of the set int ans = S . size (); return ans ; } // Driver code public static void main (String [] args) { int arr [] = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; // calculating the size of the // array int l = arr . length ; // calling the function on array int dis_elements = distinct (arr , l); System . out . print (dis_elements + "\n"); } } // This code is contributed by Rajput-Ji Python # function that accepts the array and it's size and returns # the number of distince elements def distinct (arr , l): # declaring a set container using STL S = set () for i in range (l): # inserting all elements of the # array into set S . add (arr [i]) # calculating the size of the set ans = len (S) return ans # Driver code if __name__ == '__main__' : arr = [12 , 10 , 9 , 45 , 2 , 10 , 10 , 45] # calculating the size of the array l = len (arr) # calling the function on array dis_elements = distinct (arr , l) print (dis_elements , "") # This code is contributed by Rajput-Ji C# using System ; using System.Collections.Generic ; public class GFG { // function that accepts the array and it's size and // returns the number of distince elements static int distinct (int [] arr , int len) { // declaring a set // container using STL HashSet < int > S = new HashSet < int > (); for (int i = 0 ; i < len ; i ++) { // inserting all elements of the // array into set S . Add (arr [i]); } // calculating the size of the set int ans = S . Count ; return ans ; } // Driver code public static void Main (String [] args) { int [] arr = { 12 , 10 , 9 , 45 , 2 , 10 , 10 , 45 }; // calculating the size of the array int l = arr . Length ; // calling the function on array int dis_elements = distinct (arr , l); Console . Write (dis_elements + "\n"); } } // This code is contributed by Rajput-Ji JavaScript < script > // function that accepts the array and it's size and returns // the number of distince elements function distinct (arr , len) { var S = new Set (); // declaring a set container using STL var i = 0 ; for (i = 0 ; i < len ; i ++) { S . add (arr [i]); // inserting all elements of the // array into set } var ans = S . size ; // calculating the size of the set return ans ; } // Driver code var arr = [12 , 10 , 9 , 45 , 2 , 10 , 10 , 45]; var l = arr . length ; // calculating the size of the array var dis_elements = distinct (arr , l); // calling the function on array document . write (dis_elements); // This code is contributed by Rajput-Ji. < /script> Output 5 One-liner Code - Python arr = [12 , 10 , 9 , 45 , 2 ,

10 , 10 , 45] print (len (set (arr))) JavaScript let arr = [12 , 10 , 9 , 45 , 2 , 10 , 10 , 45] console . log ((new Set (arr)). size) Output 5 Comment Article Tags: Article Tags: DSA Arrays Hash