1. Remove Duplicates Sorted Array

Problem statement:

You are given a **sorted array arr[]** containing positive integers. Your task is to **remove all duplicate elements** from this array such that each element appears only once. Return an array containing these distinct elements in the same order as they appeared.

Examples :

Input: arr[] = [2, 2, 2, 2, 2]
Output: [2]
Explanation: After removing all the duplicates only one instance of 2 will remain i.e. [2] so modified array will contains 2 at first position and you should return array containing [2] after modifying the array.

Input: arr[] = [1, 2, 4]
Output: [1, 2, 4]
Explation: As the array does not contain any duplicates so you should return [1, 2, 4].

Constraints:

$1 \leq arr.size() \leq 10^5$

$1 \leq arr[i] \leq 10^6$

CODE:

C++ (12) Start Timer

```cpp
class Solution {
  public:
    vector<int> removeDuplicates(vector<int> &arr)
    {

        if(arr.size()==0)
        {
            return {};
        }
        int j=0;
        for(int i=1; i<arr.size(); i++)
        {
            if(arr[i] != arr[j])
            {
                j++;
                arr[j] = arr[i];
            }
        }
        arr.resize(j+1);
        return arr;
    }
};
```

OUTPUT:

Output Window

Compilation Results Custom Input

Compilation Completed

• Case 1

Input:

arr[] =

2 2 2 2 2

Your Output:

[2]

Expected Output:

[2]

Koriya Leena 92410133034

Output Window

Compilation Results | Custom Input | Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 2

Accuracy : 50%

Points Scored

2 / 2

Your Total Score: 2

Time Taken

0.1

2. Smallest subarray with sum greater than x

Problem statement:

Smallest subarray with sum greater than x

Difficulty: **Easy** Accuracy: **37.07%** Submissions: **145K+** Points: **2** Average Time: **20m**

Given a number **x** and an array of integers **arr**, find the smallest subarray with sum greater than the given value. If such a subarray do not exist return 0 in that case.

Examples:

Input: x = 51, arr[] = [1, 4, 45, 6, 0, 19]
Output: 3
Explanation: Minimum length subarray is [4, 45, 6]

Input: x = 100, arr[] = [1, 10, 5, 2, 7]
Output: 0
Explanation: No subarray exist

Constraints:

$1 \leq arr.size, x \leq 10^5$

$0 \leq arr[] \leq 10^4$

CODE:

C++ (12) Start Timer

```cpp
class Solution {
public:
    int smallestSubWithSum(int x, vector<int>& arr) {
        int start = 0;
        int end = 0;
        int sum = 0;
        int minLength = arr.size() + 1;

        while (end < arr.size()) {
            int currentElement = arr[end];
            sum = sum + currentElement;

            while (sum > x) {
                int currentLength = end - start + 1;
                if (currentLength < minLength) {
                    minLength = currentLength;
                }

                int removeElement = arr[start];
                sum = sum - removeElement;
                start = start + 1;
            }

            end = end + 1;
        }

        if (minLength == arr.size() + 1) {
            return 0;
        } else {
            return minLength;
        }
    }
};
```

OUTPUT:

Output Window

Compilation Results Custom Input

Case 1

Input:

x =

51

arr[] =

1 4 45 6 0 19

Your Output:

3

Expected Output:

3

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed

1112 / 1112

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

2 / 2

Your Total Score: 4

Time Taken

0.07

3. Left most and right most index

Problem statement:

Left most and right most index

Difficulty: **Easy** Accuracy: **38.23%** Submissions: **82K+** Points: **2** Average Time: **15m**

Given a sorted array with possibly duplicate elements. The task is to find indexes of first and last occurrences of an element **X** in the given array.

Note: If the element is not present in the array **return {-1,-1} as pair**.

Example 1:

Input:
N = 9
v[] = {1, 3, 5, 5, 5, 5, 67, 123, 125}
X = 5
Output:
2 5
Explanation:
Index of first occurrence of 5 is 2
and index of last occurrence of 5 is 5.

CODE:

C++ (12) Start Timer

```cpp
class Solution {
public:
    pair<long, long> indexes(vector<long long> &arr, long long target) {
        int n = arr.size();
        long firstIndex = -1;
        long lastIndex = -1;
        int left = 0;
        int right = n - 1;
        while (left <= right) {
            int mid = (left + right) / 2;
            if (arr[mid] == target) {
                firstIndex = mid;
                right = mid - 1;}
                else if (arr[mid] < target) {
                left = mid + 1;}
                else {
                right = mid - 1;}
        }
        left = 0;
        right = n - 1;
        while (left <= right) {
            int mid = (left + right) / 2;
            if (arr[mid] == target) {
                lastIndex = mid;
                left = mid + 1;}
                else if (arr[mid] < target) {
                left = mid + 1;}
                else {
                right = mid - 1;}
        }
        return {firstIndex, lastIndex};
    }
};
```

OUTPUT:

Output Window

Compilation Results | Custom Input

Compilation Completed

Case 1

Input:

```
9
1 3 5 5 5 5 67 123 125
5
```

Your Output:

2 5

Expected Output:

2 5

Output Window

Compilation Results | Custom Input | Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed

150 / 150

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

2 / 2

Your Total Score: 6

Time Taken

0.21

4. Count BST nodes that lie in a given range

Problem statement:

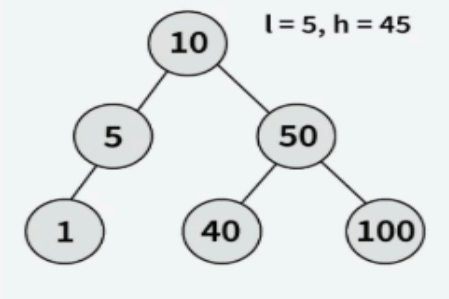
Count BST nodes that lie in a given range

Difficulty: **Medium** Accuracy: **64.84%** Submissions: **98K+** Points: **4** Average Time: **45m**

Given a Binary Search Tree (BST) and a range **l-h (inclusive)**, your task is to return the number of nodes in the BST whose value lie in the given range.

Examples :

Input: root[] = [10, 5, 50, 1, N, 40, 100], l = 5, h = 45



Output: 3
Explanation: There are three nodes in range [5, 45] = 5, 10 and 40.

CODE:

C++ (12) Start Timer

```cpp
class Solution {
public:
    int getCount(Node* root, int low, int high) {
        if (root == NULL) {
            return 0;
        }

        int count = 0;
        if (root->data >= low && root->data <= high) {
            count = 1;
        }

        int leftCount = getCount(root->left, low, high);
        int rightCount = getCount(root->right, low, high);

        return count + leftCount + rightCount;
    }
};
```

OUTPUT:

Output Window

Compilation Results Custom Input

Case 1

Input:

root[] =

10 5 50 1 N 40 100

l =

5

h =

45

Your Output:

3

Expected Output:

3

Output Window

Compilation Results Custom Input Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed

1111 / 1111

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

4 / 4

Your Total Score: 10

Time Taken

0.03

5. Longest Common Prefix of Strings

Problem statement:

Longest Common Prefix of Strings

Difficulty: **Easy** Accuracy: **29.52%** Submissions: **309K+** Points: **2** Average Time: **15m**

Given an array of strings **arr[].** Return the **longest common prefix** among each and every strings present in the array. If there's no prefix common in all the strings, return "".

Examples :

Input: arr[] = ["geeksforgeeks", "geeks", "geek", "geezer"]
Output: "gee"
Explanation: "gee" is the longest common prefix in all the given strings.

Input: arr[] = ["hello", "world"]
Output: ""
Explanation: There's no common prefix in the given strings.

Constraints:

$1 \le |arr| \le 10^3$

$1 \le |arr[i]| \le 10^3$

CODE:

C++ (12) Start Timer

```cpp
class Solution {
public:
    string longestCommonPrefix(vector<string>& arr) {
        if (arr.empty()) return "";

        string prefix = arr[0];

        for (int i = 1; i < arr.size(); i++) {
            string current = arr[i];
            int j = 0;
            while (j < prefix.length() && j < current.length() && prefix[j] == current[j]) {
                j++;
            }
            prefix = prefix.substr(0, j);
            if (prefix == "") return "";
        }

        return prefix;
    }
};
```

OUTPUT:

Output Window

Compilation Results | Custom Input

Compilation Completed

Case 1

Input:

arr [] =

geeksforgeeks geeks geek geezer

Your Output:

gee

Expected Output:

gee

Output Window

Compilation Results | Custom Input | Y.O.G.I. (AI Bot)

Problem Solved Successfully

Suggest Feedback

Test Cases Passed

1115 / 1115

Attempts : Correct / Total

1 / 1

Accuracy : 100%

Points Scored

2 / 2

Your Total Score: 12

Time Taken

0.04