

# **ARRAY QUESTION**

## **TYPES OF QUESTION**

**1 . SLIDING WINDOW (Fixed Sized, Variable Sized)**

**2. TWO – POINTER**

**3. TWO-SUM TECHNIQUE**

**4. HASHING/ HASH-MAP**

**5. PREFIX SUM**

**6. BACKTRACKING**

**7. BASIC QUESTION**

# 1. SLIDING WINDOW (Fixed Sized Window)

## 1. Max Sum Sub-array of size K

Activities Brave Web Browser May 25 8:27 PM

Max Sum Subarray of size K x + [practice.geeksforgeeks.org/problems/max-sum-subarray-of-size-k5313/1](https://practice.geeksforgeeks.org/problems/max-sum-subarray-of-size-k5313/1)

Problems Courses Get Hired Contests PTD

EG 65 🔍 📢

**Max Sum Subarray of size K**

Basic Accuracy: 49.6% Submissions: 71K+ Points: 1

Last Week Of Job Fair 2023. Explore Unlimited Opportunities Now! ↗

Given an array of integers Arr of size **N** and a number **K**. Return the maximum sum of a subarray of size **K**.

**NOTE\***: A subarray is a contiguous part of any given array.

**Example 1:**

**Input:**  
N = 4, K = 2  
Arr = [100, 200, 300, 400]

**Output:**  
700

**Explanation:**  
 $\text{Arr}_3 + \text{Arr}_4 = 700$ , which is maximum.

C++ (g++ 5.4) Start Timer

```
1 // } Driver Code Ends
2 class Solution{
3 public:
4     long maximumSumSubarray(int K, vector<int> &Arr , int N){
5         // code here
6
7         long sum =0;
8         int i=0 ,j=0;
9         long maxSum = INT_MIN;
10
11
12         while(j<N){ // perform till j reaches size of array i.e N
13             sum = sum + Arr[j];
14
15             if((j-i+1) < K){ //size of sliding window is not maintained yet
16                 j++;
17             }
18
19             else if ((j-i +1) == K){ // then maintain this sliding window and perform calculations
20                 maxSum = max( maxSum, sum);
21                 sum = sum - Arr[i];
22                 i++;
23                 j++;
24             }
25         }
26         return maxSum;
27     }
28
29     // Expected Time Complexity: O(N)
30     // Expected Auxiliary Space: O(1)
31
32
33     // Constraints:
34     // 1<=N<=106
35     // 1<=K<=N
36 };
37
38
39
40
41
42 };
43 // } Driver Code Ends
```

Custom Input Compile & Run Submit

## 2 . First negative integer in every window of size k

The screenshot shows a web browser window for the GeeksforGeeks practice platform. The URL is [practice.geeksforgeeks.org/problems/first-negative-integer-in-every-window-of-size-k3345/1](https://practice.geeksforgeeks.org/problems/first-negative-integer-in-every-window-of-size-k3345/1). The page displays a C++ code editor with a timer set to 15 minutes. The code implements a solution for finding the first negative integer in every window of size k. It uses vectors and lists to manage the windows and their elements.

**Input :**  
N = 5  
A[] = {-8, 2, 3, -6, 10}  
K = 2

**Output :**  
-8 0 -6 -6

**Explanation :**  
First negative integer for each window of size k  
{-8, 2} = -8  
{2, 3} = 0 (does not contain a negative integer)  
{3, -6} = -6  
{-6, 10} = -6

```
30 //Question : First negative integer in every window of size k
31
32 | vector<long long> printFirstNegativeInteger(long long int A[], long long int N, long long int K) {
33 |     int i=0, j=0;
34 |     list<long long> l;
35 |     vector<long long> v;
36 |
37 |     while(j < N){
38 |
39 |         //calculation : inserting -ve elements in a list
40 |         if(A[j]<0){
41 |             l.push_back(A[j]);
42 |         }
43 |
44 |         if((j-i +1) < K){
45 |             j++;
46 |         }
47 |
48 |         else if ((j-i+1 == K)){
49 |             if(l.size() == 0){
50 |                 v.push_back(0);
51 |             }
52 |             else{
53 |                 v.push_back(l.front());
54 |
55 |                 if(A[i] == l.front()){
56 |                     l.pop_front();
57 |                 }
58 |                 i++;
59 |                 j++;
60 |             }
61 |
62 |         }
63 |
64 |     }
65 |
66 |     return v;
67 |
68 | // Expected Time Complexity: O(N)
69 | // Expected Auxiliary Space: O(K)
70 |
71 }
```

Custom Input    Compile & Run    Submit

### **3. Count Occurrences of Anagrams**

<https://practice.geeksforgeeks.org/problems/count-occurrences-of-anagrams5839/1>

**Expected Time Complexity:**  $O(N)$

**Expected Auxiliary Space:**  $O(26)$  or  $O(256)$

**Constraints:**

$1 \leq |\text{pat}| \leq |\text{txt}| \leq 105$

Both string contains lowercase english letters.

## 4 . Sliding Window Maximum :

Activities    Brave Web Browser    May 25 8:45 PM

Sliding Window Maximum    +    leetcode.com/problems/sliding-window-maximum/

LeetCode    Problem List    Premium    0

Description    Editorial    Solutions (4.2K)    Submissions

### 239. Sliding Window Maximum

Hard    14.4K    468    Companies

You are given an array of integers `nums`, there is a sliding window of size `k` which is moving from the very left of the array to the very right. You can only see the `k` numbers in the window. Each time the sliding window moves right by one position.

Return the max sliding window.

**Example 1:**

```
Input: nums = [1,3,-1,-3,5,3,6,7], k = 3
Output: [3,3,5,5,6,7]
Explanation:
Window position          Max
-----          -----
[1  3  -1] -3  5  3  6  7    3
 1 [3  -1  -3] 5  3  6  7    3
 1  3 [-1  -3  5] 3  6  7    5
 1  3  -1 [-3  5  3] 6  7    5
 1  3  -1  -3 [5  3  6] 7    6
 1  3  -1  -3  5 [3  6  7]    7
```

**Example 2:**

```
Input: nums = [1], k = 1
Output: [1]
```

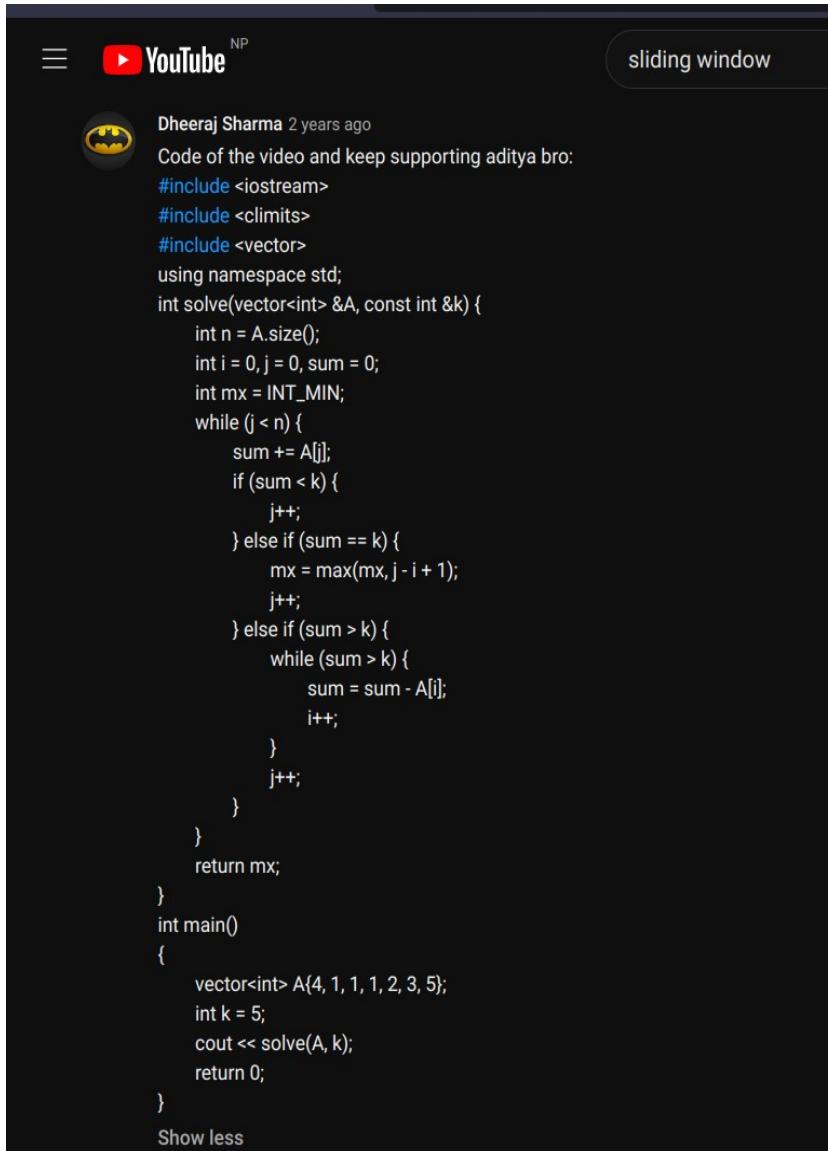
i C++    Auto

```
1 class Solution {
2 public:
3     vector<int> maxSlidingWindow(vector<int>& nums, int k) {
4         vector<int> ans;
5         int i=0;
6         int j=0;
7         list<int> l;
8         while(j<nums.size()){
9             if(l.empty()){
10                 l.push_back(nums[j]);
11             }
12             else{
13                 while(!l.empty() && l.back() < nums[j]){
14                     l.pop_back();
15                 }
16                 l.push_back(nums[j]);
17             }
18             if(j-i+1 < k){
19                 j++;
20             }
21             else if( j-i+1 == k){
22                 ans.push_back(l.front());
23                 if(nums[i]== l.front())
24                     l.pop_front();
25                 i++;
26                 j++;
27             }
28         }
29         return ans;
30     }
31 }
```

Console    Run    Submit

## 2 . SLIDING WINDOW (Variable Sized Window)

### 1. Largest Sub-array of sum K :



The image shows a YouTube video player interface. The video is titled "sliding window". The author is Dheeraj Sharma, posted 2 years ago. The video has a dark theme with a black background. The code is displayed in white text. The code implements a sliding window algorithm to find the maximum size subarray with a sum less than or equal to k.

```
#include <iostream>
#include <climits>
#include <vector>
using namespace std;
int solve(vector<int> &A, const int &k) {
    int n = A.size();
    int i = 0, j = 0, sum = 0;
    int mx = INT_MIN;
    while (j < n) {
        sum += A[j];
        if (sum < k) {
            j++;
        } else if (sum == k) {
            mx = max(mx, j - i + 1);
            j++;
        } else if (sum > k) {
            while (sum > k) {
                sum = sum - A[i];
                i++;
            }
            j++;
        }
    }
    return mx;
}
int main()
{
    vector<int> A{4, 1, 1, 1, 2, 3, 5};
    int k = 5;
    cout << solve(A, k);
    return 0;
}
```

Show less

## 2. Longest Sub-string With K Unique Characters :

The screenshot shows a web browser window for the GeeksforGeeks platform. The URL is [practice.geeksforgeeks.org/problems/longest-k-unique-characters-substring0853/1](https://practice.geeksforgeeks.org/problems/longest-k-unique-characters-substring0853/1). The page title is "Longest K unique characters substring". The problem is categorized as Medium with an accuracy of 34.65% and 74K+ submissions. A banner at the top of the page says "Last Week Of Job Fair 2023. Explore Unlimited Opportunities Now!".

**Example 1:**

**Input:**  
S = "aababebebe", K = 3  
**Output:** 7  
**Explanation:** "cbebebe" is the longest substring with K distinct characters.

**Example 2:**

**Input:**  
S = "aaaa", K = 2  
**Output:** -1

The main content area contains a C++ code editor with the following code:

```
1 // } Driver Code Ends
8 //User function template for C++
9
10 class Solution{
11 public:
12     int longestKSubstr(string str, int k) {
13         // your code here
14
15         int n = str.size();
16         int i=0, j=0, max_window_size =-1; //INT_MIN throws error ....Why ?? size cannot be less than -1 may be
17
18         unordered_map<char,int> mp;
19
20         while(j<n)
21         {
22
23             mp[str[j]]++; //Calculation to store freq of unique element in map
24
25             if(mp.size()<k)  // condition <k
26                 j++;
27
28             else if(mp.size() == k)
29             {
30                 max_window_size = max(max_window_size, (j-i+1)); // Evaluate answer from calculations
31                 j++;
32             }
33
34             else
35             {
36                 while(mp.size()>k)
37                 {
38                     mp[str[i]]--; //decrease frequency of that element
39
40                     if(mp[str[i]] == 0)
41                         mp.erase(str[i]); //erase the element whose count equals zero
42
43                     i++;
44
45                 }
46
47             }
48         }
49         j++;
50     }
51 }
```

At the bottom of the code editor, there are buttons for "Custom Input", "Compile & Run", and "Submit".

### 3. Longest Substring Without Repeating Characters :

Activities Brave Web Browser May 25 9:04 PM

Longest Substring Without Repeating Characters | Ilahi Full Video Song | Yeh Jawaani Hai Deewani

leetcode.com/problems/longest-substring-without-repeating-characters/

LeetCode Problem List Premium

Description Editorial Solutions (15.1K) Submissions

### 3. Longest Substring Without Repeating Characters

Medium 34.1K 1.5K Companies

Given a string  $s$ , find the length of the **longest substring** without repeating characters.

**Example 1:**

**Input:**  $s = "abcabcbb"$   
**Output:** 3  
**Explanation:** The answer is "abc", with the length of 3.

**Example 2:**

**Input:**  $s = "bbbbbb"$   
**Output:** 1  
**Explanation:** The answer is "b", with the length of 1.

**Example 3:**

**Input:**  $s = "pwwkew"$   
**Output:** 3  
**Explanation:** The answer is "wke", with the length of 3.  
Notice that the answer must be a substring, "pwke" is a subsequence and not a substring.

```
i C++ v . Auto
1 class Solution {
2 public:
3     int lengthOfLongestSubstring(string str) {
4
5         int n = str.size();
6         int i=0, j=0, max_window_size = 0; //INT_MIN throws error ....Why ?? size cannot be less than -1 may be
7
8         unordered_map<char,int> mp;
9
10        while(j<n)
11        {
12
13            mp[str[j]]++; //Calculation to store freq of unique element in map
14
15            if(mp.size()>(j-i+1)) // condition <k
16                j++;
17
18            else if(mp.size() == (j-i+1))
19            {
20                max_window_size = max(max_window_size, (j-i+1)); // Evaluate answer from calculations
21                j++;
22            }
23
24            else
25            {
26                while(mp.size()<(j-i+1))
27                {
28                    mp[str[i]]--; //decrease frequency of that element
29
30                    if(mp[str[i]] == 0)
31                        mp.erase(str[i]); //erase the element whose count equals zero
32                    i++;
33                }
34                j++;
35            }
36        }
37    } return max window size;
38 }
```

Console ▾ Run Submit

**4) Pick Toys : Similar to question no.2 with k=2**

**5) Minimum Window Substring : (Not Completed Yet )**

**<https://leetcode.com/problems/minimum-window-substring/>**

**See down.**

## **B. PREFIX SUM Questions**

# 1) Subarray with sum K

Youtube : <https://www.youtube.com/watch?v=xvNwoz-ufXA&t=592s>

The screenshot shows a web browser window for the GeeksforGeeks platform. The URL in the address bar is <https://practice.geeksforgeeks.org/problems/subarrays-with-sum-k-1>. The page title is "Subarrays with sum K".

**Problem Details:**

- Medium
- Accuracy: 49.74%
- Submissions: 20K+
- Points: 4

**Description:**

Given an unsorted array of integers, find the number of continuous subarrays having sum exactly equal to a given number k.

**Example 1:**

**Input:**

```
N = 5  
Arr = {10 , 2, -2, -20, 10}  
k = -10
```

**Output:** 3

**Explanation:**

Subarrays: arr[0...3], arr[1...4], arr[3..4]  
have sum exactly equal to -10.

**Code Editor:**

```
C++ (g++ 5.4) ▾ Start Timer ⏳  
1 // } Driver Code Ends  
2 //User function Template for C++  
3  
4 class Solution{  
5 public:  
6     int findSubArraySum(int arr[], int n, int k)  
7     {  
8         unordered_map<int, int> mpp;  
9  
10        int prefixSum = 0;  
11        int count = 0;  
12  
13        for(int i=0; i<n; i++){  
14            prefixSum += arr[i]; // Find prefix sum for each index it moves ahead  
15  
16            if(prefixSum == k)  
17                count++;  
18  
19            int remove = prefixSum - k;  
20  
21            count += mpp[remove]; // Reverse Eng. .... Increase the count  
22  
23            mpp[prefixSum]++; // Store the frequency of each prefixsum in map  
24  
25        }  
26  
27        return count;  
28    }  
29  
30    // Expected Time Complexity: O(NlogN)  
31    // Expected Auxiliary Space: O(N)  
32  
33    // Constraints:  
34    // 1 ≤ N ≤ 2*104  
35    // -103 ≤ Arr[i] ≤ 103  
36    // -107 ≤ k ≤ 107  
37  
38 };  
39 // } Driver Code Ends
```

Custom Input   Compile & Run  Submit

## 2. Kadane's Algorithm

Activities   Brave Web Browser   May 26 7:26 AM

Largest Sum Contiguous Suba   Kadane's Algorithm | Pract x +   practice.geeksforgeeks.org/problems/kadanes-algorithm-1587115620/1

Problems Courses Get Hired Contests ↗ POTD      65   65   🔍   📡   📈   🎵

**Kadane's Algorithm** 

Medium Accuracy: 36.28% Submissions: 692K+ Points: 4

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array **Arr[]** of **N** integers. Find the contiguous subarray (containing at least one number) which has the maximum sum and return its sum.

**Example 1:**

**Input:**  
N = 5  
Arr[] = {1,2,3,-2,5}

**Output:**  
9

**Explanation:**  
Max subarray sum is 9  
of elements (1, 2, 3, -2, 5) which is a contiguous subarray.

C++ (g++ 5.4)   Average Time: 20m   Start Timer

```
1 // } Driver Code Ends
2 class Solution{
3 public:
4     // arr: input array
5     // n: size of array
6     //Function to find the sum of contiguous subarray with maximum sum.
7     long long maxSubarraySum(int A[], int n){
8
9         // Your code here
10
11        long long sum=0;
12        long long maxSum = INT_MIN;
13
14        for(int i=0;i<n;i++){
15
16            sum += A[i];
17
18            maxSum = max(sum, maxSum);
19
20            if(sum<0)
21                sum=0;
22
23        }
24
25        return maxSum;
26    }
27
28
29
30
31
32
33
34 // Expected Time Complexity: O(N)
35 // Expected Auxiliary Space: O(1)
36
37
38 // Constraints:
39 // 1 ≤ N ≤ 106
40 // -107 ≤ A[i] ≤ 107
41 };
42 // } Driver Code Ends
```

Custom Input   Compile & Run   Submit

### 3. Range Sum Query :

Activities Brave Web Browser May 26 7:29 AM

Largest Sum Contiguous Suba Kadane's Algorithm | Practice Maximum prefix sum for a + practice.geeksforgeeks.org/problems/maximum-prefix-sum-for-a-given-range0227/1

Problems Courses Get Hired Contests POTD

GE

Average Time: 10m Start Timer

Problems Editorial Submissions Comments

#### Maximum prefix sum for a given range

Easy Accuracy: 34.88% Submissions: 2K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array of **N** integers and **Q** queries, each query having a range from index **L** to **R**. Find the maximum prefix-sum for the range **L** to **R**.

Note: Assume 0 based indexing.

**Example 1:**

**Input:**  
a[ ] = {-1, 2, 3, -5}  
Q = 2  
L<sub>1</sub> = 0, R<sub>1</sub> = 3  
L<sub>2</sub> = 1, R<sub>2</sub> = 3

**Output:**  
4 5

**Explanation:**  
The range (0, 3) in the 1st query is {-1, 2, 3, -5}, hence, the max prefix-sum will be -1 + 2 + 3 = 4. The range (1, 3) in the 2nd query is {2, 3, -5}, hence, the max prefix-sum will be 2 + 3 = 5.

```
// } Driver Code Ends
class Solution{
public:
vector<int> maxPrefixes(int a[], int L[], int R[], int N, int Q) {
    vector<int> res;
    for(int j=0; j<Q; j++){ //like For Q = 2 queries ; Q=0 --> l=2 , r=4
        // Q=1 --> l=1, r=5
        int l = L[j];
        int r = R[j];
        int ans = INT_MIN; // for comparision to get maxSum
        int sum = 0;
        for(int i=l; i<=r; i++){
            sum += a[i];
            ans = max(ans, sum);
        }
        res.push_back(ans); // return a vector that contain answersss
    }
    return res;
}
// Expected Time Complexity: O(N*Q)
// Expected Auxiliary Space: O(1)
// Constraints:
// 1 ≤ N ≤ 104
// -104 ≤ A[i]≤ 104
// 1 ≤ Q ≤ 104
// 0 ≤ L[i] ≤ R[i] < N
};

// } Driver Code Ends
```

Custom Input Compile & Run Submit

## 4. Count Subarray with Product Less than K(Sliding Window + Prefix) :

Activities   Brave Web Browser   May 26 7:33 AM

Largest Sum Contiguous Sub | Kadane's Algorithm | Practice | Maximum prefix sum for a given array | Count the subarrays having product less than k | +

practice.geeksforgeeks.org/problems/count-the-subarrays-having-product-less-than-k1708/0

Problems Courses Get Hired Contests ↗ POTD

Editorial Submissions Comments

### Count the subarrays having product less than k

Hard Accuracy: 21.0% Submissions: 49K+ Points: 8

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array of positive numbers, the task is to find the number of possible contiguous subarrays having product less than a given number k.

**Example 1:**

**Input :**  
n = 4, k = 10  
a[] = {1, 2, 3, 4}

**Output :**  
7

**Explanation:**  
The contiguous subarrays are {1}, {2}, {3}, {4} {1, 2}, {1, 2, 3} and {2, 3} whose count is 7.

C++ (g++ 5.4) Start Timer

```
1 // } Driver Code Ends
2
3
4
5
6
7
8
9
10 class Solution{
11 public:
12     int countSubArrayProductLessThanK(const vector<int>& a, int n, long long k) {
13         int i = 0, j = 0;
14         long long prod = 1;
15         int count = 0;
16
17         while (j < n) {
18             prod = prod * a[j];
19
20             while (i <= j && prod >= k) {
21                 prod = prod / a[i];
22                 i++;
23             }
24
25             //measure how many sub array exist
26             int len = j - i + 1;
27             count = count + len;
28             j++;
29         }
30
31         return count;
32     }
33
34
35
36     // Expected Time Complexity: O(n)
37     // Expected Auxiliary Space: O(1)
38
39
40
41     // Constraints:
42     // 1<=n<=10^5
43     // 1<=k<=10^15
44     // 1<=a[i]<=10^5
45 };
46 // } Driver Code Ends
```

Custom Input Compile & Run Submit

## 5. Minimum Size Subarray Sum:(Sliding Window + Prefix Sum) :

Activities 📺 Brave Web Browser May 26 10:55 AM

Ultimate Bollywood Fusion 20 Mere Ghar Ram Aaye Hai [S] Smallest subarray with sum x +

practice.geeksforgeeks.org/problems/smallest-subarray-with-sum-greater-than-x5651/1

Problems Courses Get Hired Contests ↗ POTD

GeeksforGeeks

Average Time: 20m Start Timer

Smallest subarray with sum greater than x

Easy Accuracy: 37.07% Submissions: 93K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array of integers ( $A[]$ ) and a number  $x$ , find the smallest subarray with sum greater than the given value. If such a subarray does not exist return 0 in that case.

**Note:** The answer always exists. It is guaranteed that  $x$  doesn't exceed the summation of  $a[i]$  (from 1 to N).

**Example 1:**

**Input:**  
 $A[] = \{1, 4, 45, 6, 0, 19\}$   
 $x = 51$

**Output:** 3

**Explanation:**  
Minimum length subarray is  
 $\{4, 45, 6\}$

```
int smallestSubWithSum(int arr[], int n, int x)
{
    // Your code goes here

    // Initialize sum and minimum length
    int sum=0;
    int min_len = n+1;

    // Initialize starting and ending indexes
    int i=0, j=0;
    while (j < n) {

        // Keep adding array elements while sum
        // is smaller than or equal to x

        while(sum <= x && j < n) {
            sum = sum +arr[j];
            j++;
        }

        // If sum becomes greater than x.
        while (sum > x && i < n) {

            // Update minimum length if needed
            if (j - i < min_len)
                min_len = j-i;

            // remove starting elements
            sum =sum- arr[i];
            i++;
        }
    }
    return min_len;
}
// Expected Time Complexity: O(N)
// Expected Auxiliary Space: O(1)

// Constraints:
// 1 ≤ N, x ≤ 10^5
// 1 ≤ A[] ≤ 10^4
```

Custom Input Compile & Run Submit

**(NOT COMPLETED YET)**

**6. Counting Subarray with score Less Than K :**

<https://leetcode.com/problems/count-subarrays-with-score-less-than-k/>

**7. Maximum Subarray Sum Circular :**

<https://www.geeksforgeeks.org/maximum-contiguous-circular-sum/>

**8. Largest Subarray with 0 Sum:**

<https://www.geeksforgeeks.org/find-the-largest-subarray-with-0-sum/>

**9. Subarray Sum Divisible by K :**

<https://www.geeksforgeeks.org/count-sub-arrays-sum-divisible-k/>

**10. Smallest Subarray with All Occurrences of a Most Frequent Element:**

<https://www.geeksforgeeks.org/smallest-subarray-with-all-occurrences-of-a-most-frequent-element/>

## **C. HASHING / HASHMAP Questions**

**Continued ....**

# 1. Find the element that appears once :

The screenshot shows a web browser window with the following details:

- Address Bar:** practice.geeksforgeeks.org/problems/element-appearing-once2552/0
- Page Title:** Find the element that appears once
- Header:** Activities, Brave Web Browser, May 26 11:21 AM
- Page Content:**
  - Problem Section:** Problems, Courses, Get Hired, Contests, POTD.
  - Problem Details:** Find the element that appears once, Medium, Accuracy: 53.69%, Submissions: 63K+, Points: 4.
  - Call-to-Action:** Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!
  - Description:** Given a sorted array A[] of N positive integers having all the numbers occurring exactly twice, except for one number which will occur only once. Find the number occurring only once.
  - Example 1:** Input: N = 5, A = {1, 1, 2, 5, 5}. Output: 2. Explanation: Since 2 occurs once, while other numbers occur twice, 2 is the answer.
  - Example 2:** (Partially visible)
- Code Editor:** C++ (g++ 5.4) code editor with the following content:

```
1 // } Driver Code Ends
2 //User function template for C++
3 class Solution{
4 public:
5     int search(int arr[], int n){
6         //code
7
8         unordered_map <int , int> mp;
9
10        for(int i=0; i<n; i++){ // Insert elements to map with its frequency
11            mp[arr[i]]++;
12        }
13
14        for(auto it=mp.begin() ; it!=mp.end(); it++){ // it returns an iterator
15            if( mp[it->first] == 1 ) // check freq of that element
16                return (it->first); // if freq ==1 return that as an answer
17        }
18
19    }
20
21    // Expected Time Complexity: O(Log(N)).
22    // Expected Auxiliary Space: O(1).
23
24    // Constraints
25    // 0 < N <= 10^6
26    // 0 <= A[i] <= 10^9
27    //
28    // } Driver Code Ends
```

Buttons at the bottom of the code editor include: Custom Input, Compile & Run, and Submit.

## 2. Check if two arrays are equal or not :

The screenshot shows a web browser window with the URL [practice.geeksforgeeks.org/problems/check-if-two-arrays-are-equal-or-not3847/0](https://practice.geeksforgeeks.org/problems/check-if-two-arrays-are-equal-or-not3847/0). The page title is "Check if two arrays are equal or not". The browser's address bar also lists other tabs like "Smallest subarray with all occurences of a character". The browser interface includes a top bar with user info, a toolbar, and a sidebar with navigation links.

**Problem Details:**

- Name:** Check if two arrays are equal or not
- Category:** Basic
- Accuracy:** 42.18%
- Submissions:** 255K+
- Points:** 1

**Description:**

Given two arrays **A** and **B** of equal size **N**, the task is to find if given arrays are equal or not. Two arrays are said to be equal if both of them contain same set of elements, arrangements (or permutation) of elements may be different though.

**Note:** If there are repetitions, then counts of repeated elements must also be same for two array to be equal.

**Example 1:**

**Input:**

```
N = 5
A[] = {1,2,5,4,0}
B[] = {2,4,5,0,1}
```

**Output:** 1

**Explanation:** Both the array can be rearranged to {0,1,2,4,5}

**Code Editor:**

```
1 // } Driver Code Ends
2 //User function template for C++
3
4 class Solution{
5 public:
6
7     //Function to check if two arrays are equal or not.
8     bool check(vector<ll> A, vector<ll> B, int n) {
9         //code here
10
11         unordered_map <int,int> mp1, mp2;
12
13         for(int it : A){
14             mp1[it]++;
15         }
16
17         for(int it : B){
18             mp2[it]++;
19         }
20
21         //compare the frequency
22         for(auto it = mp1.begin(); it != mp1.end(); it++){ //Traversing map mp1
23
24             //If frequency not same return false
25             if(mp2[it->first] != it->second){
26                 return 0;
27             }
28         }
29         return 1;
30     }
31
32     // Expected Time Complexity : O(N)
33     // Expected Auxilliary Space : O(N)
34
35     // Constraints:
36     // 1<=N<=10^7
37     // 1<=A[],B[]<=10^18
38     // };
39
40     // } Driver Code Ends
```

**Buttons:** Custom Input, Compile & Run, Submit

### 3. Twice Counter (Count the number of words that appear exactly twice in the list)

Activities Brave Web Browser May 26 11:28 AM

Twice Counter | Practice

practice.geeksforgeeks.org/problems/twice-counter4236/0

Problems Courses Get Hired Contests POTD

OG

Start Timer

C++ (g++ 5.4) // } Driver Code Ends

```
1 // } Driver Code Ends
2
3 class Solution
4 {
5     public:
6         int countWords(string list[], int n)
7     {
8         //code here.
9         unordered_map<string, int> mp;
10
11         for(int i=0; i<n; i++){
12             mp[list[i]]++; //Insert elements inside map mp
13         }
14
15         int counter =0;
16
17         for(auto &pr : mp){ // IN key-value , if value == 2 of any key increase counter
18             if(pr.second == 2){
19                 counter++;
20             }
21         }
22
23         return counter;
24     }
25
26
27
28
29
30 // Expected Time Complexity: O(N)
31 // Expected Auxiliary Space: O(N)
32
33
34 // Constraints:
35 // 1<= N <= 104
36 };
37 // } Driver Code Ends
```

67 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37

Custom Input Compile & Run Submit

Easy Accuracy: 62.61% Submissions: 34K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given a list of N words. Count the number of words that appear exactly twice in the list.

**Example 1:**

**Input:**  
N = 3  
list = {Geeks, For, Geeks}

**Output:** 1

**Explanation:** 'Geeks' is the only word that appears twice.

**Example 2:**

**Input:**  
N = 8  
list = {Tom, Jerry, Thomas, Tom, Jerry,

## **D. TWO POINTER Questions**

**Continued ....**

# 1. Remove duplicate elements from sorted Array :

Activities Brave Web Browser May 26 11:39 AM

Remove duplicate element x + practice.geeksforgeeks.org/problems/remove-duplicate-elements-from-sorted-array/1

Problems Courses Get Hired Contests PTD

67 🔍 📢

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given a sorted array **A[]** of size **N**, delete all the duplicated elements from **A[]**. Modify the array such that if there are **X** distinct elements in it then the first **X** positions of the array should be filled with them in increasing order and return the number of distinct elements in the array.

**Note:**

1. Don't use set or HashMap to solve the problem.
2. You must return the number of distinct elements(**X**) in the array, the generated output will print all the elements of the modified array from index 0 to **X**-1.

**Example 1:**

**Input:**  
N = 5  
Array = {2, 2, 2, 2, 2}

**Output:** {2}

**Explanation:** After removing all the duplicates only one instance of 2 will remain.

C++ (g++ 5.4) Average Time: 20m Start Timer

```
1 // } Driver Code Ends
2 //User function template for C++
3
4 // Question : Remove duplicate elements from sorted Array
5
6 class Solution{
7 public:
8     int remove_duplicate(int a[],int n){
9
10     // code here
11     int i =0;
12     for(int j =1; j<n; j++){
13         if(a[i]== a[j]){
14             i++;
15             a[i] = a[j];
16         }
17     }
18     return i+1;
19 }
20
21
22
23
24
25
26
27
28
29
30
31
32 // Expected Time Complexity: O(N)
33 // Expected Auxiliary Space: O(1)
34
35
36 // Constraints:
37 // 1 ≤ N ≤ 104
38 // 1 ≤ A[i] ≤ 106
39
40
41 };
42 // } Driver Code Ends
```

Custom Input Compile & Run Submit

## 2. Quick Left Rotation by 1 place :

Activities    Brave Web Browser    May 26 11:41 AM

Quick Left Rotation | Practice    +    practice.geeksforgeeks.org/problems/quick-left-rotation3806/1

Problems   Courses   Get Hired   Contests   POTD

GE

**Quick Left Rotation**

Basic   Accuracy: 34.84%   Submissions: 51K+   Points: 1

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array **arr[]** of size **N** and an integer **K**, the task is to left rotate the array **K** indexes

**Example 1:**

**Input:** N = 7, K = 2  
arr[] = {1, 2, 3, 4, 5, 6, 7}  
**Output:** 3 4 5 6 7 1 2  
**Explanation:** Rotation of the above array by 2 will make the output array .

**Example 2:**

**Input:** N = 6, K = 12  
arr[] = {1, 2, 3, 4, 5, 6}  
**Output:** 1 2 3 4 5 6

```
1 // } Driver Code Ends
2 class Solution{
3
4     public:
5         void leftRotate(int arr[], int k, int n)
6     {
7         // Your code goes here
8
9         k=k%n;
10
11         int temp[k]; //Array size of k
12
13         for(int i=0; i<k; i++)
14             temp[i]=arr[i];
15
16         for(int i=k; i<n; i++)
17             arr[i-k] =arr[i];
18
19         int j=0;
20         for(int i=n-k; i<n; i++){
21             arr[i]=temp[j];
22             j++;
23         }
24
25         // Expected Time Complexity: O(N).
26         // Expected Auxiliary Space: O(1).
27
28         // Constraints:
29         // 1 ≤ N ≤ 105
30         // 1 ≤ K ≤ 105
31         // -100 ≤ arr[i] ≤ 100
32
33     }
34
35     // } Driver Code Ends
```

Custom Input    Compile & Run    Submit

### 3. Rotating an array by D-places :

Activities Brave Web Browser

May 26 11:43 AM

Rotating an Array | Practice

practice.geeksforgeeks.org/problems/reversal-algorithm5340/1

Problems Courses Get Hired Contests POTD

GE

Average Time: 20m Your Time: 0m 21s

67 7 67

Problems Editorial Submissions Comments

#### Rotating an Array

Basic Accuracy: 44.48% Submissions: 90K+ Points: 1

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array of size **N**. The task is to rotate array by **D** elements where **D ≤ N**.

**Example 1:**

**Input:**  
N = 7  
Arr[] = {1, 2, 3, 4, 5, 6, 7}  
D = 2  
**Output:** 3 4 5 6 7 1 2

**Explanation:**  
Rotate by 1: [2, 3, 4, 5, 6, 7, 1]  
Rotate by 2: [3, 4, 5, 6, 7, 1, 2]

C++ (g++ 5.4) // Driver Code Ends //User function template for C++ class Solution{ public: void leftRotate(int arr[], int n, int d) { // code here d =d%n; int temp[d]; //array of size d for(int i=0; i<d; i++) { temp[i] =arr[i]; } for(int i=d; i<n; i++){ arr[i-d]=arr[i]; } int j=0; for(int i=n-d; i<n; i++){ arr[i]=temp[j]; j++; } } // Expected Time Complexity: O(N) // Expected Auxiliary Space: O(1) // Constraints: // 1 ≤ N ≤ 105 // 1 ≤ Arr[i] ≤ 1000 // 0 ≤ D ≤ N }; // Driver Code Ends

Custom Input Compile & Run Submit

## 4. Move all zeroes to end of array :

Activities Brave Web Browser

Move all zeroes to end of array +

May 26 11:46 AM

practice.geeksforgeeks.org/problems/move-all-zeroes-to-end-of-array0751/1

Problems Courses Get Hired Contests POTD

GeeksforGeeks

C++ (g++ 5.4) Average Time: 15m Start Timer

Move all zeroes to end of array

Easy Accuracy: 45.51% Submissions: 101K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array arr[] of N positive integers. Push all the zeros of the given array to the right end of the array while maintaining the order of non-zero elements.

**Example 1:**

**Input:**  
N = 5  
Arr[] = {3, 5, 0, 0, 4}

**Output:** 3 5 4 0 0

**Explanation:** The non-zero elements preserve their order while the 0 elements are moved to the right.

**Example 2:**

```
1 // } Driver Code Ends
2 //User function template for C++
3 class Solution{
4 public:
5     void pushZeroesToEnd(int arr[], int n) {
6         // code here
7         int j = -1;           // pick the 1st index containing zero as element in variable j //
8         for(int i=0; i<n; i++){
9             if(arr[i]==0){
10                 j = i;
11                 break;
12             }
13         }
14
15         if(j== -1){ //handles case if you dont have any zero element in an array
16             exit;
17         }
18
19         else{
20             int temp; //Assisted variable for swapping
21             for(int i=j+1; i<n; i++){
22                 if(arr[i]!=0){           // if ith element is not zero, swap it with jth element(i.e element
23                                         //containing zero)
24                     temp=arr[i];
25                     arr[i]=arr[j];
26                     arr[j]=temp;
27                     j++;
28                 }
29             }
30         }
31     }
32
33     // Expected Time Complexity: O(N)
34     // Expected Auxiliary Space: O(1)
35
36     // Constraints:
37     // 1 ≤ N ≤ 10^5
38     // 0 ≤ arr[i] ≤ 10^5
39
40 }
41
42 };
43
44 // } Driver Code Ends
```

Custom Input Compile & Run Submit

## **E. BASIC Questions**

**Continued ....**

# 1. Largest Element in Array :

The screenshot shows a web browser window for the 'Brave Web Browser' with the URL [practice.geeksforgeeks.org/problems/largest-element-in-array4009/0](https://practice.geeksforgeeks.org/problems/largest-element-in-array4009/0). The page displays a C++ solution for finding the largest element in an array.

**Problem Details:**

- Problem: Largest Element in Array
- Category: Basic
- Accuracy: 67.48%
- Submissions: 175K+
- Points: 1

**Description:**

Given an array  $A[]$  of size  $n$ . The task is to find the largest element in it.

**Example 1:**

**Input:**  
 $n = 5$   
 $A[] = \{1, 8, 7, 56, 90\}$

**Output:**  
90

**Explanation:**  
The largest element of given array is 90.

**Code Snippet (C++ 5.4):**

```
1 // } Driver Code Ends
2 //User function Template for C++
3
4 class Solution {
5 public:
6     int largest(vector<int> &arr, int n)
7     {
8         int largest = arr[0];
9         for(int i=0; i<n; i++){
10             if (arr[i]>largest){
11                 largest = arr[i];
12             }
13         }
14         return largest;
15     }
16
17 // Expected Time Complexity: O(N)
18 // Expected Auxiliary Space: O(1)
19
20 // Constraints:
21 // 1 <= n <= 103
22 // 0 <= A[i] <= 103
23 // Array may contain duplicate elements.
24
25 };
26
27 // } Driver Code Ends
```

**Buttons at the bottom:**

- Custom Input
- Compile & Run
- Submit

## 2. Second Largest Element in An Array :

Activities Brave Web Browser

May 26 11:59 AM

Second Largest | Practice

practice.geeksforgeeks.org/problems/second-largest3735/1

Problems Courses Get Hired Contests POTD

67 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39

Second Largest

School Accuracy: 26.72% Submissions: 302K+ Points: 0

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array **Arr** of size **N**, print second largest distinct element from an array.

**Example 1:**

**Input:**  
N = 6  
Arr[] = {12, 35, 1, 10, 34, 1}  
**Output:** 34

**Explanation:** The largest element of the array is 35 and the second largest element is 34.

**Code:**

```
C++ (g++ 5.4) Average Time: 15m Your Time: 2m 9s
```

```
1 // } Driver Code Ends
2 //User function template for C++
3 class Solution{
4 public:
5     // Function returns the second
6     // largest elements
7
8     int print2largest(int arr[], int n) {
9         // code here
10        int largest = arr[0];
11        int slargest = -1;
12
13        for (int i=0; i<n; i++){
14            if(arr[i]>largest){
15                slargest = largest;
16                largest = arr[i];
17            }
18            else if(arr[i]<largest && arr[i]>slargest){
19                slargest= arr[i];
20            }
21        }
22
23        return slargest;
24    }
25
26
27
28
29
30
31
32 // Expected Time Complexity: O(N)
33 // Expected Auxiliary Space: O(1)
34
35 // Constraints:
36 // 2 ≤ N ≤ 10^5
37 // 1 ≤ Arr[i] ≤ 10^5
38 }
39 // } Driver Code Ends
```

Custom Input Compile & Run Submit

### 3. Check if array is sorted :

Activities Brave Web Browser May 26 12:00 PM

Check if array is sorted | PRACTICE practice.geeksforgeeks.org/problems/check-if-an-array-is-sorted0701/1

Problems Courses Get Hired Contests POTD

OG Average Time: 15m Start Timer

Problem Editorial Submissions Comments

#### Check if array is sorted

Easy Accuracy: 39.37% Submissions: 123K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array `arr[]` of size `N`, check if it is sorted in non-decreasing order or not.

**Example 1:**

**Input:**  
`N = 5`  
`arr[] = {10, 20, 30, 40, 50}`

**Output:** 1

**Explanation:** The given array is sorted.

**Example 2:**

**Input:**  
`N = 6`  
`arr[] = {90, 80, 100, 70, 40, 30}`

**Output:** 0

```
1 // [ Driver Code Starts
8 // User function template for C++
9
10 class Solution {
11 public:
12     bool arraySortedOrNot(int arr[], int n) {
13         // code here
14         for (int i=1; i<n; i++){
15             if(arr[i]>=arr[i-1]){
16                 //then think xa dont do anything
17             }
18             else
19                 return 0;
20         }
21         return 1;
22     }
23
24 // Expected Time Complexity: O(N)
25 // Expected Auxiliary Space: O(1)
26
27
28 // Constraints:
29 // 1 ≤ N ≤ 10^5
30 // 1 ≤ Arr[i] ≤ 10^6
31 };
32 // [ Driver Code Ends]
```

Custom Input Compile & Run Submit

## 4. Missing number in Contiguous Array :

Activities Brave Web Browser May 26 12:02 PM

Missing number | Practice + practice.geeksforgeeks.org/problems/missing-number4257/1

Problems Courses Get Hired Contests POTD

67 7 11 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 32 33 34 35 36 37 38 39 40 41 42 43 44 45 46 47 48 49 50 51 52 53 54 55 56 57 58 59 60 61 62 63 64

int missingNumber(int A[], int N){  
 // Your code goes here  
 int sum=N\*(N+1)/2;  
 int s1=0;  
  
 for(int i=0; i<N-1 ; i++){  
 s1 = s1 + A[i];  
 }  
 return sum-s1;  
 /\*  
 \*Another Optimal Solution  
 \*/  
  
 int xor1 =0,xor2 =0;  
 int n = N-1;  
  
 for(int i=0; i<n; i++){  
 xor2 = xor2 ^ a[i];  
 xor1 = xor1 ^ (i+1);  
 }  
  
 xor1 = xor1 ^ N;  
  
 return xor1 ^ xor2;  
}

Custom Input Compile & Run Submit

### Missing number

Basic Accuracy: 45.15% Submissions: 55K+ Points: 1

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Vaibhav likes to play with numbers and he has **N** numbers. One day he was placing the numbers on the playing board just to count that how many numbers he has. He was placing the numbers in increasing order i.e. from **1** to **N**. But when he was putting the numbers back into his bag, some numbers fell down onto the floor. He picked up all the numbers but one number, he couldn't find. Now he has to go somewhere urgently, so he asks you to find the missing number.

**NOTE:** Don't use Sorting

#### Example 1:

**Input:**  
N = 4  
A[] = {1, 4, 3}

**Output:**  
2

## 5. Maximum Consecutive Ones :

Activities   Brave Web Browser   May 26 12:04 PM

Missing number | Practice | Ge Max Consecutive Ones - Leet + leetcode.com/problems/max-consecutive-ones/ Problem List Premium

LeetCode

Description   Editorial   Solutions (4.1K)   Submissions

485. Max Consecutive Ones   Hint   ⓘ

Easy   ✓ 4.4K   429   ⚡   Companies

Given a binary array `nums`, return the maximum number of consecutive 1's in the array.

Example 1:

**Input:** `nums = [1,1,0,1,1,1]`  
**Output:** 3  
**Explanation:** The first two digits or the last three digits are consecutive 1s. The maximum number of consecutive 1s is 3.

Example 2:

**Input:** `nums = [1,0,1,1,0,1]`  
**Output:** 2

Constraints:

- `1 <= nums.length <= 105`
- `nums[i]` is either 0 or 1.

i C++   Auto

```
1 class Solution {
2 public:
3     int findMaxConsecutiveOnes(vector<int>& nums) {
4         int maxi = 0;
5         int counter = 0;
6
7         for(int i=0; i<nums.size(); i++){
8             if(nums[i]==1){
9                 counter++;
10                maxi = max(maxi , counter);
11            }
12            else{
13                counter = 0 ;
14            }
15        }
16
17        return maxi;
18    }
19 }
```

Console ^

Run   Submit

# 6. Union of Two Sorted Arrays :

Activities Brave Web Browser May 26 12:06 PM

Missing number | Practice | Ge Union of Two Sorted Array +

Brave Web Browser

practice.geeksforgeeks.org/problems/union-of-two-sorted-arrays-1587115621/1

Problems Courses Get Hired Contests ↴ POTD

GeeksforGeeks

Average Time: 20m Start Timer

Problem Editorial Submissions Comments

### Union of Two Sorted Arrays

Easy Accuracy: 31.39% Submissions: 126K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Union of two arrays can be defined as the common and distinct elements in the two arrays.

Given two sorted arrays of size **n** and **m** respectively, find their union.

**Example 1:**

**Input:**  
n = 5, arr1[] = {1, 2, 3, 4, 5}  
m = 3, arr2 [] = {1, 2, 3}

**Output:** 1 2 3 4 5

**Explanation:** Distinct elements including both the arrays are: 1 2 3 4 5.

**Example 2:**

```
C++ (g++ 5.4) ▾
9 //arr1,arr2 : the arrays
10 // n, m: size of arrays
11 //Function to return a list containing the union of the two arrays.
12 vector<int> findUnion(int arr1[], int arr2[], int n, int m)
13 {
14     //return vector with correct order of elements
15     //Expected Time Complexity: O(n+m).
16     //Expected Auxiliary Space: O(n+m)
17
18     vector <int> unionArr;
19     int i=0;
20     int j=0;
21     while(i<n && j <m){ //implies if in both arrays i and j are somewhere pointing
22         if(arr1[i] <= arr2[j]){
23             if(unionArr.size() == 0 || unionArr.back() != arr1[i]){
24                 unionArr.push_back(arr1[i]);
25             }
26             i++;
27         }
28         else{
29             if(unionArr.size() == 0 || unionArr.back() != arr2[j]){
30                 unionArr.push_back(arr2[j]);
31             }
32             j++;
33         }
34     }
35
36     while( i<n ){ //implies arr2 sakiyo tara some elements are still left in arr1.. traverse arr1 now
37         if(unionArr.size() == 0 || unionArr.back() != arr1[i]){
38             unionArr.push_back(arr1[i]);
39         }
40         i++;
41     }
42
43     while( j<m ){ //implies arr1 sakiyo but still some elements left in arr2 ..traverse arr2 now
44         if(unionArr.size() == 0 || unionArr.back() != arr2[j]){
45             unionArr.push_back(arr2[j]);
46         }
47         j++;
48     }
49
50     return unionArr;
51 }
```

Custom Input Compile & Run Submit

## **F. TWO-Sum Technique**

**Continued ....**

# 1.Two Sum:

Activities Brave Web Browser May 27 9:59 AM

Key Pair | Practice | GeeksF x + practice.geeksforgeeks.org/problems/key-pair5616/1

Problems Courses Get Hired Contests ↗ POTD

GeeksforGeeks

78 78 78

Problem Editorial Submissions Comments

Average Time: 20m Start Timer

Key Pair

Easy Accuracy: 30.61% Submissions: 182K+ Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array **Arr** of **N** positive integers and another number **X**. Determine whether or not there exist two elements in **Arr** whose sum is exactly **X**.

**Example 1:**

**Input:**  
N = 6, X = 16  
Arr[] = {1, 4, 45, 6, 10, 8}

**Output:** Yes

**Explanation:** Arr[3] + Arr[4] = 6 + 10 = 16

**Example 2:**

**Input:**  
N = 5, X = 10  
Arr[] = {1, 2, 4, 3, 6}

```
C++ (g++ 5.4) ▾ Average Time: 20m Start Timer
```

```
9 class Solution{
10 public:
11     bool hasArrayTwoCandidates(int arr[], int n, int x) {
12         // code here
13         int l=0, r=n-1;
14
15         //Sort the array
16         sort(arr, arr + n);
17
18         while(l<r){
19             if(arr[l] + arr[r] == x)
20                 return 1;
21             else if ( arr[l] + arr[r] <x){
22                 l++;
23             }
24             else
25                 r--;
26         }
27         return 0;
28     }
29
30 // Let an array be {1, 4, 45, 6, 10, -8} and sum to find be 16
31 // After sorting the array
32 // A = {-8, 1, 4, 6, 10, 45}
33
34 // Initialize l = 0, r = 5
35 // A[l] + A[r] ( -8 + 45) > 16 => decrement r. Now r = 4
36 // A[l] + A[r] ( -8 + 10) increment l. Now l = 1
37 // A[l] + A[r] ( 1 + 10) increment l. Now l = 2
38 // A[l] + A[r] ( 4 + 10) increment l. Now l = 3
39 // A[l] + A[r] ( 6 + 10) == 16 => Found candidates (return 1)
40
41
42 // Expected Time Complexity: O(N)
43 // Expected Auxiliary Space: O(N)
44
45 // Constraints:
46 // 1 ≤ N ≤ 10^5
47 // 1 ≤ Arr[i] ≤ 10^5
```

Custom Input Compile & Run Submit

# 1. Find Triplet with Sum = Zero:

Activities Brave Web Browser May 27 10:03 AM

Find triplets with zero sum + practice.geeksforgeeks.org/problems/find-triplets-with-zero-sum/1

Problems Courses Get Hired Contests POTD

78 🔍 📡

### Find triplets with zero sum

Basic Accuracy: 25.81% Submissions: 224K+ Points: 1

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an array `arr[]` of `n` integers. Check whether it contains a triplet that sums up to zero.

**Example 1:**

**Input:** `n = 5, arr[] = {0, -1, 2, -3, 1}`  
**Output:** 1  
**Explanation:** 0, -1 and 1 forms a triplet with sum equal to 0.

**Example 2:**

**Input:** `n = 3, arr[] = {1, 2, 3}`  
**Output:** 0  
**Explanation:** No triplet with zero sum exists.

C++ (g++ 5.4) Average Time: 20m Start Timer

```
11 bool findTriplets(int arr[], int n)
12 {
13     bool found = false;
14
15     // sort array elements
16     sort(arr, arr + n);
17
18     for (int i = 0; i < n - 1; i++) {
19         // initialize left and right
20         int l = i + 1;
21         int r = n - 1;
22         int x = arr[i];
23
24         while (l < r) {
25             if (x + arr[l] + arr[r] == 0) {
26                 // return 1 if its sum is zero
27                 found = true;
28                 return 1;
29                 // break;
30             }
31
32             // If sum of three elements is less
33             // than zero then increment in left
34             else if (x + arr[l] + arr[r] < 0)
35                 l++;
36
37             // if sum is greater than zero then
38             // decrement in right side
39             else
40                 r--;
41         }
42     }
43     if (found == false)
44         return 0;
45 }
46 // Constraints
47 // Expected Time Complexity: O(n^2)      // 1 <= n <= 10^4
48 // Expected Auxiliary Space: O(1)          // -10^6 <= A[i] <= 10^6
49 // Driver Code Ends
```

Custom Input Compile & Run Submit

### 3. Subarray with given sum:

Activities Leafpad May 27 10:07 AM

Subarray with given sum | + practice.geeksforgeeks.org/problems/subarray-with-given-sum-1587115621/1

Problems Courses Get Hired Contests POTD

Subarray with given sum

Easy Accuracy: 16.5% Submissions: 1.1M Points: 2

Join the most popular course on DSA. Master Skills & Become Employable by enrolling today!

Given an unsorted array **A** of size **N** that contains only positive integers, find a continuous sub-array that adds to a given number **S** and return the left and right index(**1-based indexing**) of that subarray.

In case of multiple subarrays, return the subarray indexes which come first on moving from left to right.

**Note:-** You have to return an ArrayList consisting of two elements left and right. In case no such subarray exists return an array consisting of element -1.

**Example 1:**

**Input:**  
N = 5, S = 12  
A[] = {1,2,3,7,5}

**Output:** 2 4

**Explanation:** The sum of elements

C++ (g++ 5.4) Average Time: 20m Start Timer

```
vector<int> subarraySum(vector<int>arr, int n, long long s)
{
    int i = 0;
    int j = 0;
    long long sum = 0;
    vector<int> v;

    while (j < n)
    {
        if (s == 0){
            v.push_back(-1);
            return v;
        }
        sum = sum + arr[j]; //calculation

        if (sum < s){
            j++;
        }

        else{
            while (sum > s){ // eliminate from beginnig while 's' becomes less or equal to sum
                sum -= arr[i];
                i++;
            }

            // if sum becomes equal to s then execute belows code , else j++ , go to line 25
            if (sum == s){
                v.push_back(i + 1);
                v.push_back(j + 1);
                return v;
            }
            j++;
        }
    }
    v.push_back(-1);
    return v;
}
```

\*(Untitled)

File Edit Search Options Help

Expected Time Complexity: O(N)  
Expected Auxiliary Space: O(1)

Constraints:  
1 <= N <= 10^5  
1 <= A[i] <= 10^9  
0 <= S <= 10^14

Custom Input Compile & Run Submit

## 4. Count pairs with given sum :

The screenshot shows a web browser window for the GeeksforGeeks practice platform. The URL is [practice.geeksforgeeks.org/problems/count-pairs-with-given-sum5022/1](https://practice.geeksforgeeks.org/problems/count-pairs-with-given-sum5022/1). The page title is "Count pairs with given sum".

**Problem Details:**

- Difficulty: Easy
- Accuracy: 31.49%
- Submissions: 299K+
- Points: 2

**Description:**

Given an array of **N** integers, and an integer **K**, find the number of pairs of elements in the array whose sum is equal to **K**.

**Example 1:**

**Input:**  
N = 4, K = 6  
arr[] = {1, 5, 7, 1}

**Output:** 2

**Explanation:**  
arr[0] + arr[1] = 1 + 5 = 6  
and arr[1] + arr[3] = 5 + 1 = 6.

**Code Editor:**

```
C++ (g++ 5.4) Average Time: 20m Start Timer
1 // } Driver Code Ends
2 //User function template for C++
3
4 class Solution{
5 public:
6     int getPairsCount(int arr[], int n, int k) {
7         // code here
8
9         unordered_map<int, int> mp;
10        int count = 0;
11
12        for (int i = 0; i < n; i++) {
13            int temp = k - arr[i];
14            count = count + mp[temp]; // Increment count by the frequency of temp
15
16            mp[arr[i]]++; // Update frequency for the current element by inserting new
17                           // element to map
18        }
19
20    }
21
22    // Expected Time Complexity: O(N)
23    // Expected Auxiliary Space: O(N)
24
25    // Constraints:
26    // 1 <= N <= 10^5
27    // 1 <= K <= 10^8
28    // 1 <= Arr[i] <= 10^6
29
30    // } Driver Code Ends
```

Custom Input Compile & Run Submit