DSA DAY 8 21/11/24

1.[**16. 3Sum Closest**](https://leetcode.com/problems/3sum-closest/)

Solved

Medium

Topics

Companies

Given an integer array nums of length n and an integer target, find three integers in nums such that the sum is closest to target.

Return *the sum of the three integers*.

You may assume that each input would have exactly one solution.

**Example 1:**

**Input:** nums = [-1,2,1,-4], target = 1

**Output:** 2

**Explanation:** The sum that is closest to the target is 2. (-1 + 2 + 1 = 2).

**Example 2:**

**Input:** nums = [0,0,0], target = 1

**Output:** 0

**Explanation:** The sum that is closest to the target is 0. (0 + 0 + 0 = 0).

**Constraints:**

* 3 <= nums.length <= 500
* -1000 <= nums[i] <= 1000
* -104 <= target <= 104

Code:

class Solution {

public:

    int threeSumClosest(vector<int>& nums, int target) {

       sort(nums.begin(),nums.end());

       int n=nums.size();

       int ans=INT\_MAX;

       int s1;

       for(int i=0;i<n-1;i++){

            int l=i+1;

            int r=n-1;

            while(l<r){

                int s=nums[i]+nums[r]+nums[l];

                if(abs(s-target)<ans){

                    ans=abs(s-target);

                    s1=s;

                }

                if(s<target){

                    l+=1;

                }

                else{

                    r-=1;

                }

            }

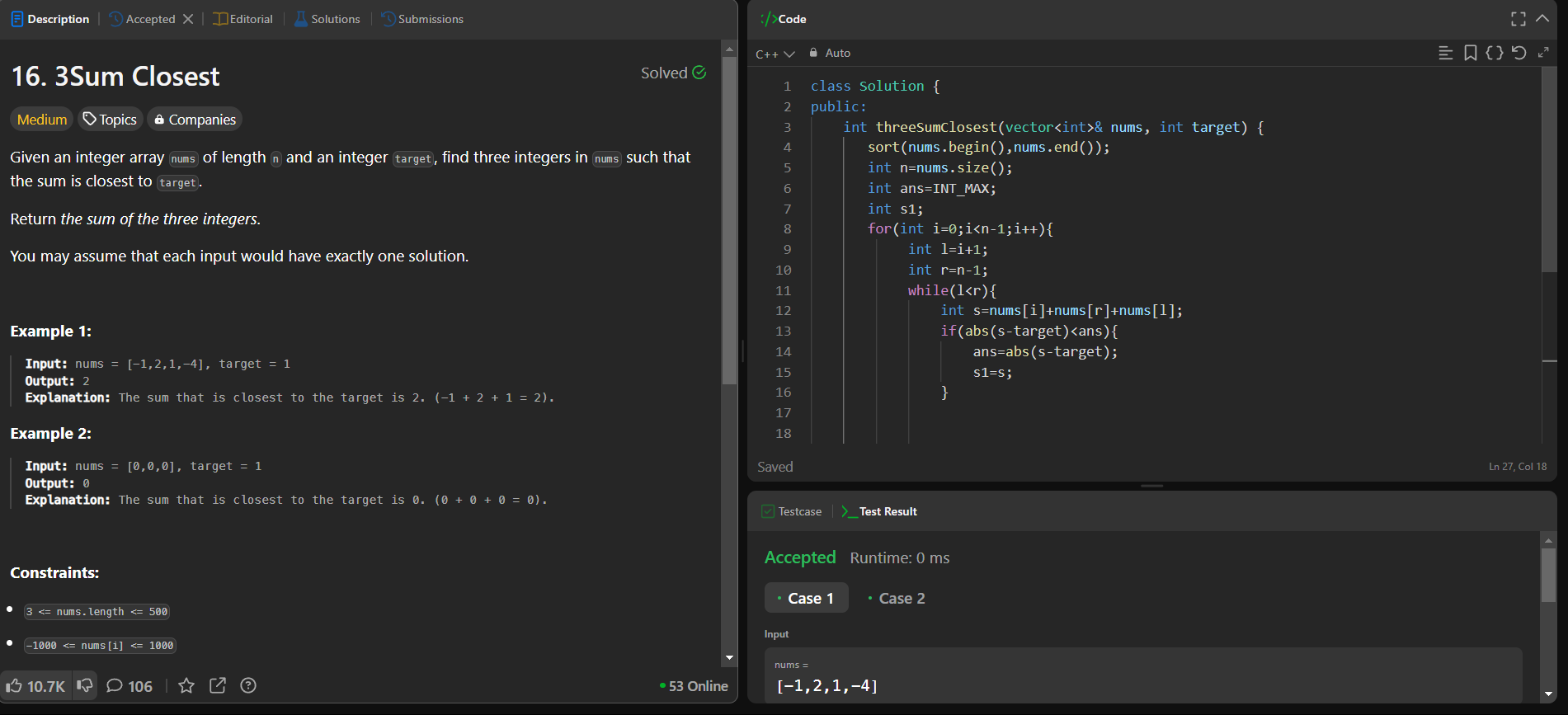
        }

        return s1;

    }

};

Output:



2.[**45. Jump Game II**](https://leetcode.com/problems/jump-game-ii/)

Solved

Medium

Topics

Companies

You are given a **0-indexed** array of integers nums of length n. You are initially positioned at nums[0].

Each element nums[i] represents the maximum length of a forward jump from index i. In other words, if you are at nums[i], you can jump to any nums[i + j] where:

* 0 <= j <= nums[i] and
* i + j < n

Return *the minimum number of jumps to reach*nums[n - 1]. The test cases are generated such that you can reach nums[n - 1].

**Example 1:**

**Input:** nums = [2,3,1,1,4]

**Output:** 2

**Explanation:** The minimum number of jumps to reach the last index is 2. Jump 1 step from index 0 to 1, then 3 steps to the last index.

**Example 2:**

**Input:** nums = [2,3,0,1,4]

**Output:** 2

Code:

class Solution {

public:

    int jump(vector<int>& nums) {

        int s=0;

        int n=nums.size();

        int f=0;

        int e=0;

        for(int i=0;i<n-1;i++){

            f=max(f,i+nums[i]);

            if(f>=n-1){

                s++;

                break;

            }

            if(i==e){

                s++;

                e=f;

            }

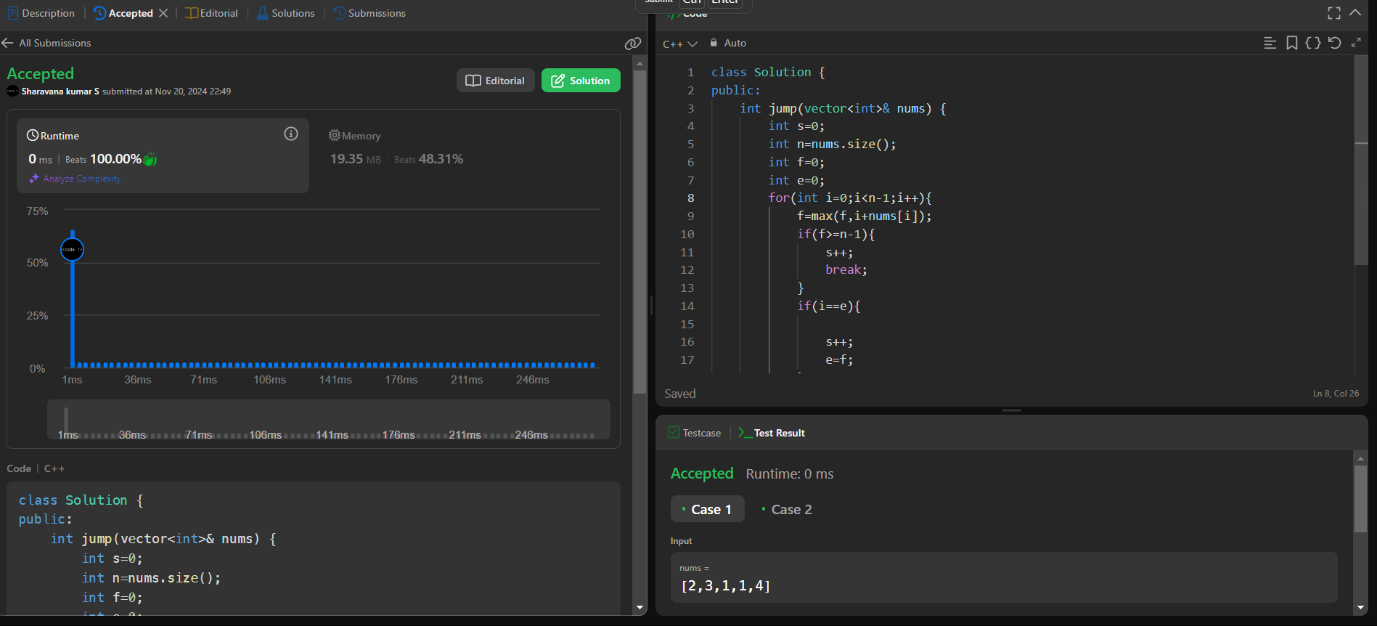
        }

        return s;

    }

};

Output:



3.[**49. Group Anagrams**](https://leetcode.com/problems/group-anagrams/)

Solved

Medium

Topics

Companies

Given an array of strings strs, group the

anagrams

 together. You can return the answer in **any order**.

**Example 1:**

**Input:** strs = ["eat","tea","tan","ate","nat","bat"]

**Output:** [["bat"],["nat","tan"],["ate","eat","tea"]]

**Explanation:**

* There is no string in strs that can be rearranged to form "bat".
* The strings "nat" and "tan" are anagrams as they can be rearranged to form each other.
* The strings "ate", "eat", and "tea" are anagrams as they can be rearranged to form each other.

**Example 2:**

**Input:** strs = [""]

**Output:** [[""]]

**Example 3:**

**Input:** strs = ["a"]

**Output:** [["a"]]

**Constraints:**

* 1 <= strs.length <= 104
* 0 <= strs[i].length <= 100
* strs[i] consists of lowercase English letters.

Code:

class Solution {

public:

    vector<vector<string>> groupAnagrams(vector<string>& strs) {

        vector<vector<string>> s;

        map<string,vector<string>> m;

        for(int i=0;i<strs.size();i++){

            string s1=strs[i];

            sort(s1.begin(),s1.end());

            m[s1].push\_back(strs[i]);

        }

        for(auto i:m){

            s.push\_back(i.second);

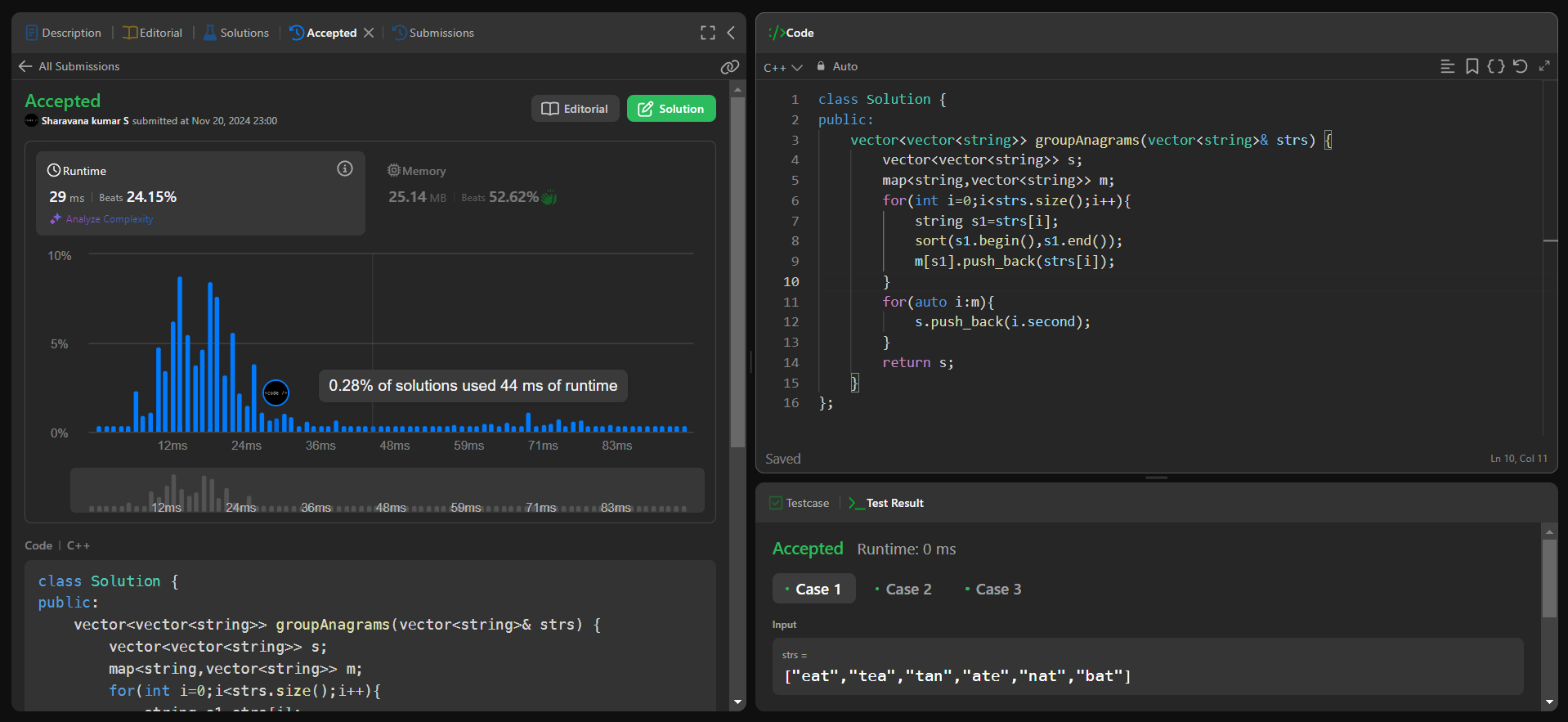
        }

        return s;

    }

};

Output:



4.[**122. Best Time to Buy and Sell Stock II**](https://leetcode.com/problems/best-time-to-buy-and-sell-stock-ii/)

Solved

Medium

Topics

Companies

You are given an integer array prices where prices[i] is the price of a given stock on the ith day.

On each day, you may decide to buy and/or sell the stock. You can only hold **at most one** share of the stock at any time. However, you can buy it then immediately sell it on the **same day**.

Find and return *the****maximum****profit you can achieve*.

**Example 1:**

**Input:** prices = [7,1,5,3,6,4]

**Output:** 7

**Explanation:** Buy on day 2 (price = 1) and sell on day 3 (price = 5), profit = 5-1 = 4.

Then buy on day 4 (price = 3) and sell on day 5 (price = 6), profit = 6-3 = 3.

Total profit is 4 + 3 = 7.

**Example 2:**

**Input:** prices = [1,2,3,4,5]

**Output:** 4

**Explanation:** Buy on day 1 (price = 1) and sell on day 5 (price = 5), profit = 5-1 = 4.

Total profit is 4.

**Example 3:**

**Input:** prices = [7,6,4,3,1]

**Output:** 0

**Explanation:** There is no way to make a positive profit, so we never buy the stock to achieve the maximum profit of 0.

**Constraints:**

* 1 <= prices.length <= 3 \* 104
* 0 <= prices[i] <= 104

Code:

class Solution {

public:

    int maxProfit(vector<int>& prices) {

        int p=0;

        int i=0;

        while(i<prices.size()-1){

            if(prices[i+1]>prices[i]){

                p+=prices[i+1]-prices[i];

            }

            i+=1;

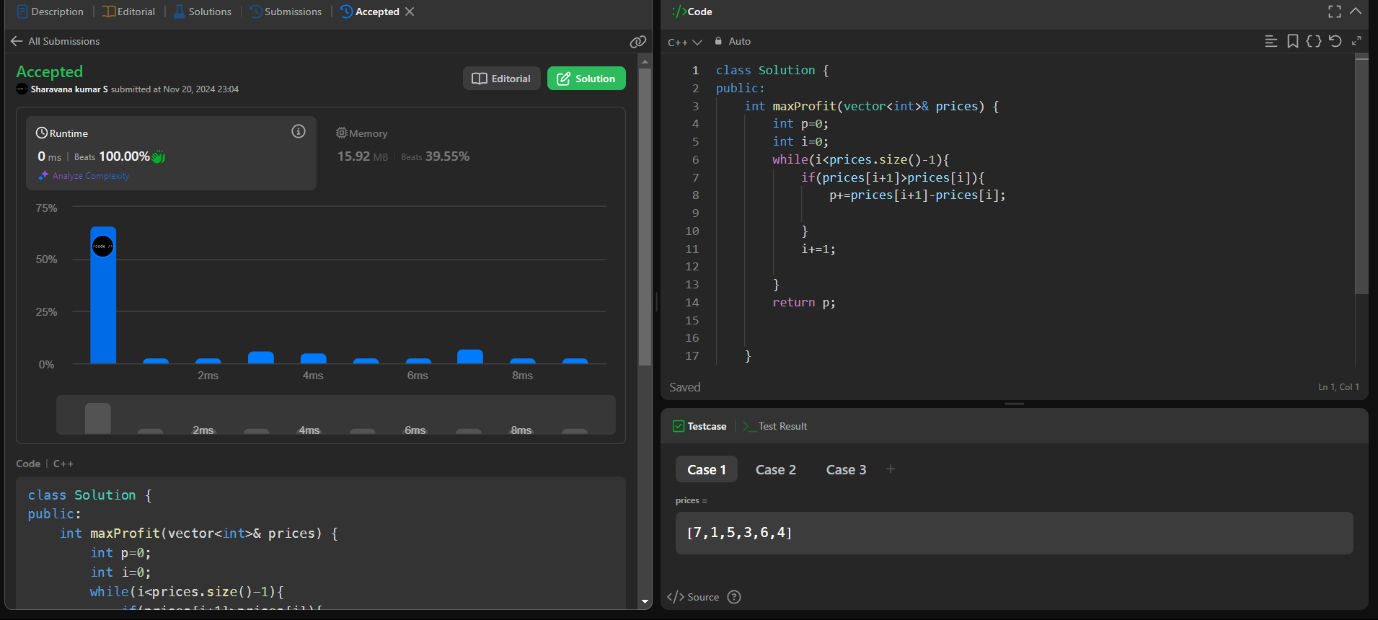
        }

        return p;

    }

};

Output:



9. **Searching an element in a sorted array (Ternary Search)**

Difficulty: **Medium**Accuracy: **50.2%**Submissions: **13K+**Points: **4**

Given a sorted array **arr[]** of size **N** and an integer **K**. The task is to check if K is present in the array or not using ternary search.  
  
[**Ternary Search**](http://www.geeksforgeeks.org/ternary-search/)- It is a divide and conquer algorithm that can be used to find an element in an array. In this algorithm, we divide the given array into three parts and determine which has the key (searched element).

**Example 1:**

**Input:**

N = 5, K = 6

arr[] = {1,2,3,4,6}

**Output:** 1

**Exlpanation:** Since, 6 is present in

the array at index 4 (0-based indexing),

output is 1.

**Example 2:**

**Input:**

N = 5, K = 2

arr[] = {1,3,4,5,6}

**Output:** -1

**Exlpanation:** Since, 2 is not present

in the array, output is -1.

**Your Task:**  
You don't need to read input or print anything. Complete the function **ternarySearch()** which takes the sorted array **arr[]**, its size **N** and the element **K** as input parameters and returns**1** if K is present in the array, else it returns **-1**.

**Expected Time Complexity:**O(Log3N)  
**Expected Auxiliary Space:**O(1)

**Constraints:**  
1 < N < 106  
1 < K < 106  
1 < arr[i] < 106

Try more examples

Code:

//{ Driver Code Starts

#include <bits/stdc++.h>

using namespace std;

// } Driver Code Ends

class Solution{

public:

// Function to find element in sorted array

// arr: input array

// N: size of array

// K: element to be searche

int ternarySearch(int arr[], int N, int K)

{

// Your code here

int l=0;

int r=N-1;

while(l<=r){

int m1 = l + (r - l) / 3;

int m2 = r - (r - l) / 3;

if(arr[m1]==K){

return 1;

}

if(arr[m2]==K){

return 1;

}

else if(K<arr[m1]){

r=m1-1;

}

else if(K>arr[m2]){

l=m2+1;

}

else{

l=m1+1;

r=m2-1;

}

}

return -1;

}

};

//{ Driver Code Starts.

int main(void)

{

int t;

cin >> t;

while(t--){

int N, K;

cin >> N >> K;

int arr[N];

for(int i = 0;i<N;i++){

cin >> arr[i];

}

Solution ob;

cout << ob.ternarySearch(arr, N, K) << endl;

cout << "~" << "\n";

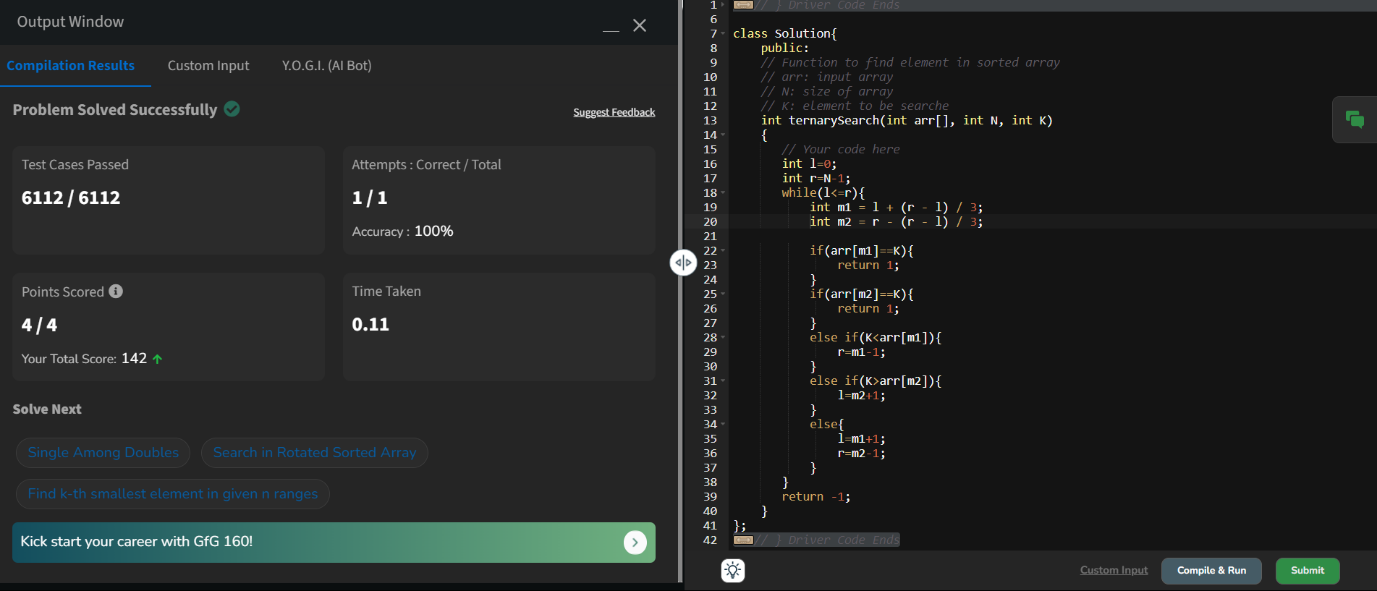
}

return 0;

}

// } Driver Code Ends

Output:



10,Interpolation Search

Code:

class Solution {

public:

int search(vector<int>& nums, int target) {

int n=nums.size();

int low=0;

int high=n-1;

while(low<=high && target>=nums[low] && target<=nums[high] ){

int p = low + ((double)(high - low) / (nums[high] - nums[low])) \* (target - nums[low]);

if(nums[p]==target){

return p;

}

if(nums[p]<target){

low=p+1;

}

else{

high=p-1;

}

}

return -1;

}

};

