DSA PRACTICE – 9

Name: Deepak S

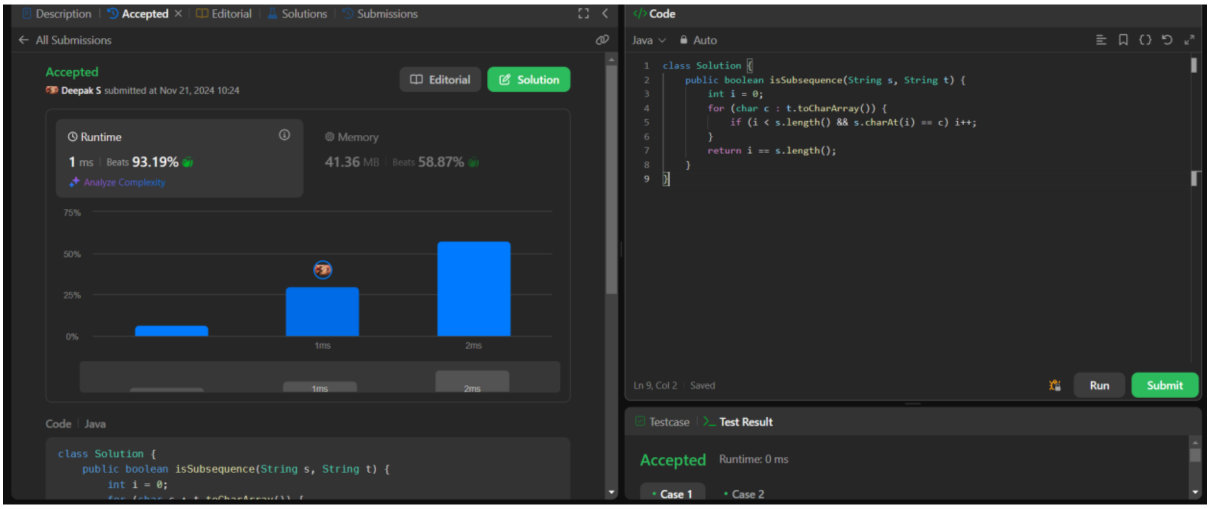
Register No: 22IT018

1.Is Subsequence

Code:

```java
class Solution {
    public boolean isSubsequence(String s, String t) {
        int i = 0;
        for (char c : t.toCharArray()) {
            if (i < s.length() && s.charAt(i) == c) i++;
        }
        return i == s.length();
    }
}
```

Output:



Accepted
Deepak S submitted at Nov 21, 2024 10:24

Runtime
1 ms | Beats 93.19%

Memory
41.36 MB | Beats 58.87%

Code | Java

```java
class Solution {
    public boolean isSubsequence(String s, String t) {
        int i = 0;
```

```java
class Solution {
    public boolean isSubsequence(String s, String t) {
        int i = 0;
        for (char c : t.toCharArray()) {
            if (i < s.length() && s.charAt(i) == c) i++;
        }
        return i == s.length();
    }
}
```

Ln 9, Col 2 | Saved

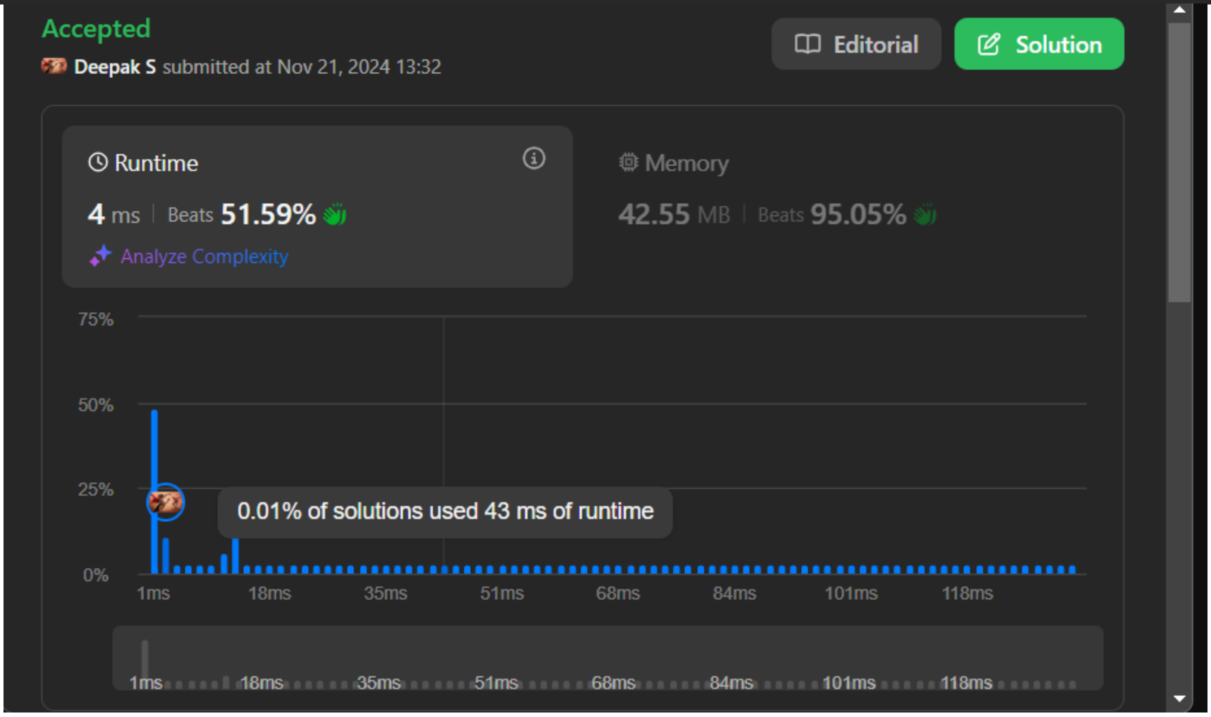
Accepted Runtime: 0 ms

2.Valid Palindrome

Code:

```
class Solution {
    public boolean isPalindrome(String s) {
        s = s.toLowerCase();
        int i = 0;
        int j = s.length()-1;
        while(i < j){
            if(!Character.isDigit(s.charAt(i)) &&
!Character.isLetter(s.charAt(i))){
                i++;
                continue;
            }
            if(!Character.isDigit(s.charAt(j)) &&
!Character.isLetter(s.charAt(j))){
                j--;
                continue;
            }
            if(!(s.charAt(i) == s.charAt(j))){
                return false;
            }
            i++;
            j--;
        }
        return true;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 13:32

Runtime

4 ms | Beats 51.59%

Memory

42.55 MB | Beats 95.05%

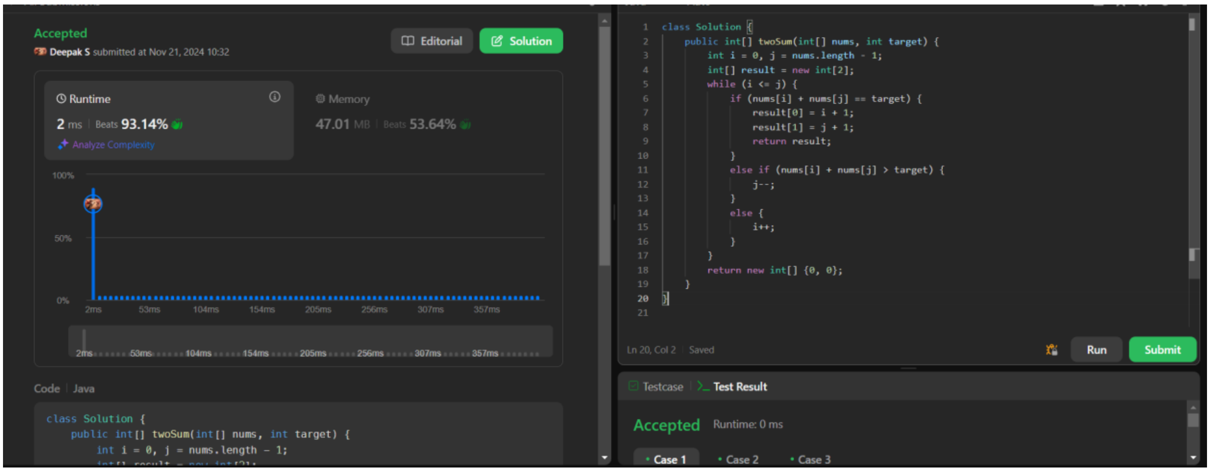
0.01% of solutions used 43 ms of runtime

3.Two Sum II - Input Array Is Sorted

Code:

```java
class Solution {
    public int[] twoSum(int[] nums, int target) {
        int i = 0, j = nums.length - 1;
        int[] result = new int[2];
        while (i <= j) {
            if (nums[i] + nums[j] == target) {
                result[0] = i + 1;
                result[1] = j + 1;
                return result;
            }
            else if (nums[i] + nums[j] > target) {
                j--;
            }
            else {
                i++;
            }
        }
        return new int[] {0, 0};
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 10:32

Runtime

2 ms | Beats 93.14%

Memory

47.01 MB | Beats 53.64%

Code | Java

```java
class Solution {
    public int[] twoSum(int[] nums, int target) {
        int i = 0, j = nums.length - 1;
```

```java
class Solution {
    public int[] twoSum(int[] nums, int target) {
        int i = 0, j = nums.length - 1;
        int[] result = new int[2];
        while (i <= j) {
            if (nums[i] + nums[j] == target) {
                result[0] = i + 1;
                result[1] = j + 1;
                return result;
            }
            else if (nums[i] + nums[j] > target) {
                j--;
            }
            else {
                i++;
            }
        }
        return new int[] {0, 0};
    }
}
```

Ln 20, Col 2 | Saved

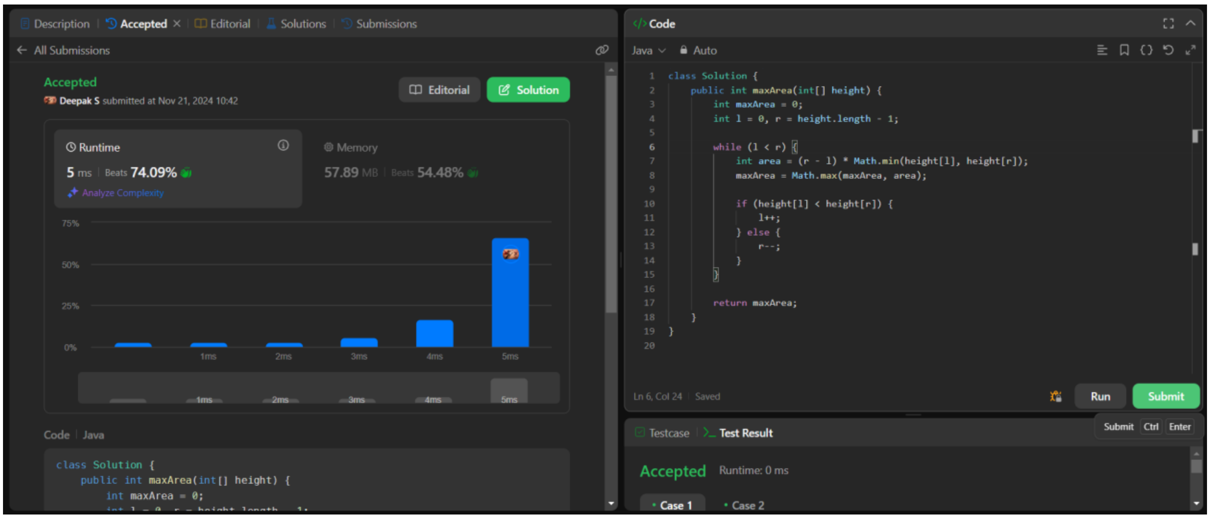
Accepted Runtime: 0 ms

4.Container With Most Water

Code:

```java
class Solution {
    public int maxArea(int[] height) {
        int maxArea = 0;
        int l = 0, r = height.length - 1;

        while (l < r) {
            int area = (r - l) * Math.min(height[l], height[r]);
            maxArea = Math.max(maxArea, area);

            if (height[l] < height[r]) {
                l++;
            } else {
                r--;
            }
        }

        return maxArea;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 10:42

Runtime: 5 ms | Beats 74.09%

Memory: 57.89 MB | Beats 54.48%

Code | Java

```java
class Solution {
    public int maxArea(int[] height) {
        int maxArea = 0;
```

```java
class Solution {
    public int maxArea(int[] height) {
        int maxArea = 0;
        int l = 0, r = height.length - 1;

        while (l < r) {
            int area = (r - l) * Math.min(height[l], height[r]);
            maxArea = Math.max(maxArea, area);

            if (height[l] < height[r]) {
                l++;
            } else {
                r--;
            }
        }

        return maxArea;
    }
}
```

Ln 6, Col 24 | Saved

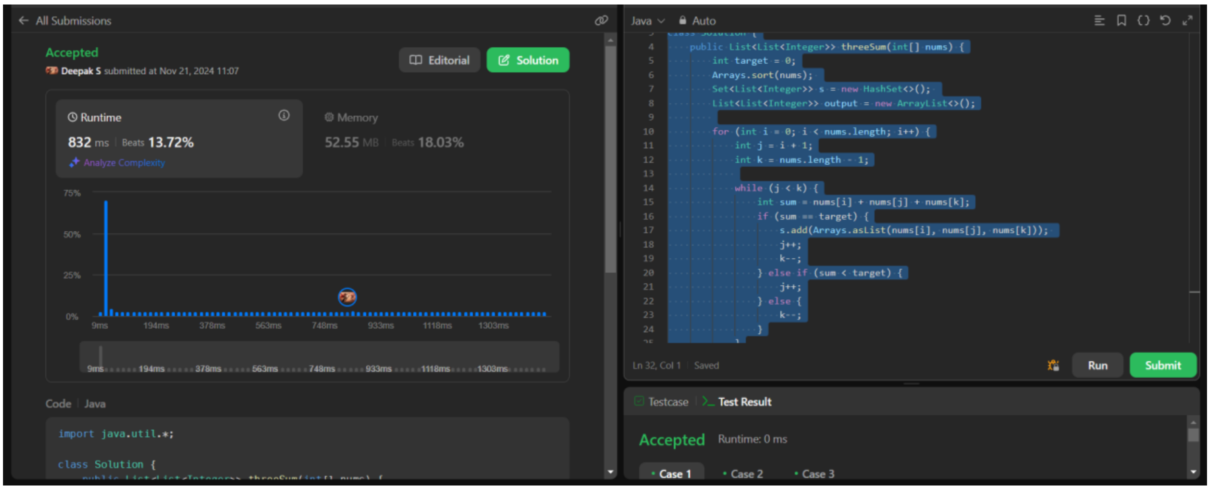
Accepted Runtime: 0 ms

5.3Sum

Code:

```
import java.util.*;

class Solution {
    public List<List<Integer>> threeSum(int[] nums) {
        int target = 0;
        Arrays.sort(nums);
        Set<List<Integer>> s = new HashSet<>();
        List<List<Integer>> output = new ArrayList<>();

        for (int i = 0; i < nums.length; i++) {
            int j = i + 1;
            int k = nums.length - 1;

            while (j < k) {
                int sum = nums[i] + nums[j] + nums[k];
                if (sum == target) {
                    s.add(Arrays.asList(nums[i], nums[j], nums[k]));
                    j++;
                    k--;
                } else if (sum < target) {
                    j++;
                } else {
                    k--;
                }
            }
        }

        output.addAll(s);
        return output;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 11:07

Runtime

832 ms | Beats 13.72%

Analyze Complexity

Memory

52.55 MB | Beats 18.03%

Code | Java

```
import java.util.*;

class Solution {
```

```
    public List<List<Integer>> threeSum(int[] nums) {
        int target = 0;
        Arrays.sort(nums);
        Set<List<Integer>> s = new HashSet<>();
        List<List<Integer>> output = new ArrayList<>();

        for (int i = 0; i < nums.length; i++) {
            int j = i + 1;
            int k = nums.length - 1;

            while (j < k) {
                int sum = nums[i] + nums[j] + nums[k];
                if (sum == target) {
                    s.add(Arrays.asList(nums[i], nums[j], nums[k]));
                    j++;
                    k--;
                } else if (sum < target) {
                    j++;
                } else {
                    k--;
                }
```

Ln 32, Col 1 | Saved

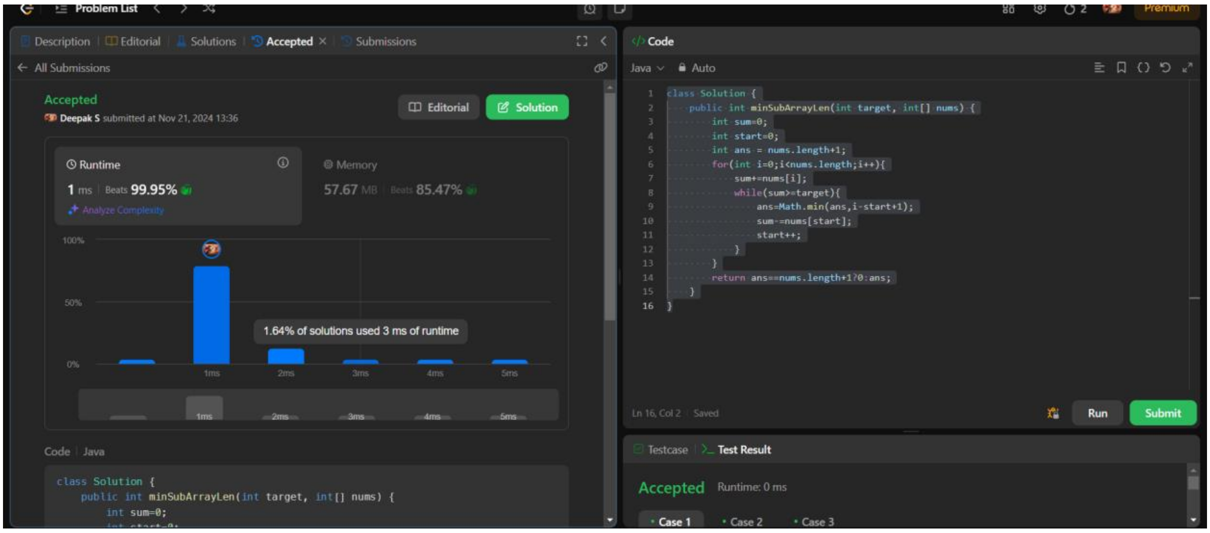
Accepted Runtime: 0 ms

6.Minimum Size Subarray Sum

Code:

```
class Solution {
    public int minSubArrayLen(int target, int[] nums) {
        int sum=0;
        int start=0;
        int ans = nums.length+1;
        for(int i=0;i<nums.length;i++){
            sum+=nums[i];
            while(sum>=target){
                ans=Math.min(ans,i-start+1);
                sum-=nums[start];
                start++;
            }
        }
        return ans==nums.length+1?0:ans;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 13:36

Runtime
1 ms | Beats 99.95%

Memory
57.67 MB | Beats 85.47%

1.64% of solutions used 3 ms of runtime

Code | Java

```
class Solution {
    public int minSubArrayLen(int target, int[] nums) {
        int sum=0;
```

```
class Solution {
    public int minSubArrayLen(int target, int[] nums) {
        int sum=0;
        int start=0;
        int ans = nums.length+1;
        for(int i=0;i<nums.length;i++){
            sum+=nums[i];
            while(sum>=target){
                ans=Math.min(ans,i-start+1);
                sum-=nums[start];
                start++;
            }
        }
        return ans==nums.length+1?0:ans;
    }
}
```

Ln 16, Col 2 | Saved

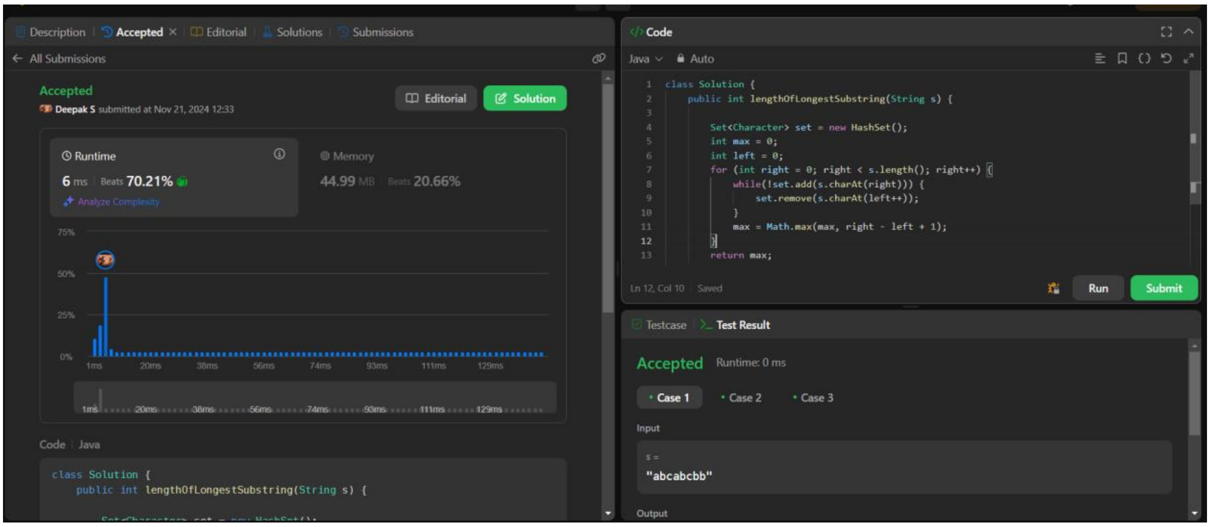
Accepted Runtime: 0 ms

7.Longest Substring Without Repeating Characters

Code:

```java
class Solution {
    public int lengthOfLongestSubstring(String s) {

        Set<Character> set = new HashSet();
        int max = 0;
        int left = 0;
        for (int right = 0; right < s.length(); right++) {
            while(!set.add(s.charAt(right))) {
                set.remove(s.charAt(left++));
            }
            max = Math.max(max, right - left + 1);
        }
        return max;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 12:33

Runtime 6 ms | Beats 70.21%

Memory 44.99 MB | Beats 20.66%

Code | Java

```java
class Solution {
    public int lengthOfLongestSubstring(String s) {

        Set<Character> set = new HashSet();
        int max = 0;
        int left = 0;
        for (int right = 0; right < s.length(); right++) {
            while(!set.add(s.charAt(right))) {
                set.remove(s.charAt(left++));
            }
            max = Math.max(max, right - left + 1);
        }
        return max;
```

Ln 12, Col 10 | Saved

Accepted Runtime: 0 ms

Input

```
s =
"abcabcbb"
```

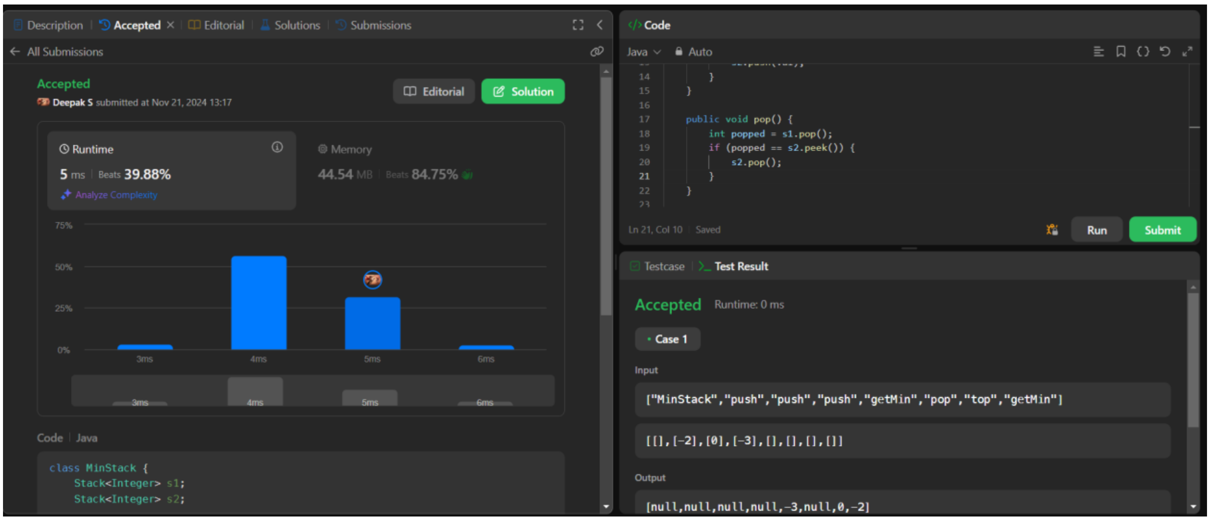
Output

8.Min Stack

Code:

```
class MinStack {
    Stack<Integer> s1;
    Stack<Integer> s2;

    public MinStack() {
        s1 = new Stack<>();
        s2 = new Stack<>();
    }

    public void push(int val) {
        s1.push(val);
        if (s2.isEmpty() || val <= s2.peek()) {
            s2.push(val);
        }
    }

    public void pop() {
        int popped = s1.pop();
        if (popped == s2.peek()) {
            s2.pop();
        }
    }

    public int top() {
        return s1.peek();
    }

    public int getMin() {
        return s2.peek();
    }
}


/**
 * Your MinStack object will be instantiated and called as such:
 * MinStack obj = new MinStack();
 * obj.push(val);
 * obj.pop();
 * int param_3 = obj.top();
 * int param_4 = obj.getMin();
 */
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 13:17

Runtime

5 ms | Beats 39.88%

Memory

44.54 MB | Beats 84.75%

Code | Java

```
class MinStack {
    Stack<Integer> s1;
    Stack<Integer> s2;
```

```
        }
    }

    public void pop() {
        int popped = s1.pop();
        if (popped == s2.peek()) {
            s2.pop();
        }
    }
```

Ln 21, Col 10 | Saved

Accepted Runtime: 0 ms

• Case 1

Input

```
["MinStack","push","push","push","getMin","pop","top","getMin"]
```

```
[[],[-2],[0],[-3],[],[],[],[]]
```

Output

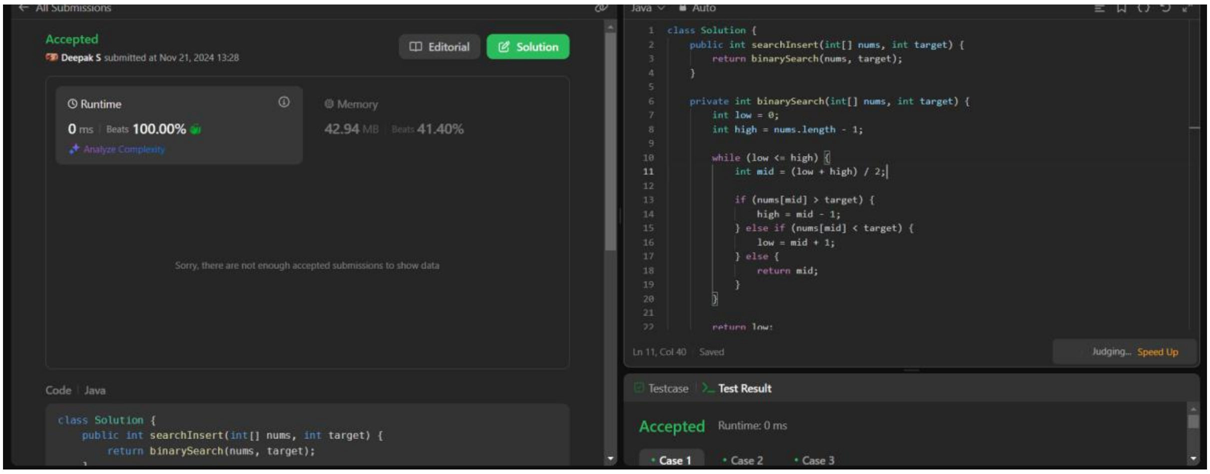
```
[null,null,null,null,-3,null,0,-2]
```

9.Search Insert Position0

Code:

```java
class Solution {
    public int searchInsert(int[] nums, int target) {
        return binarySearch(nums, target);
    }

    private int binarySearch(int[] nums, int target) {
        int low = 0;
        int high = nums.length - 1;

        while (low <= high) {
            int mid = (low + high) / 2;

            if (nums[mid] > target) {
                high = mid - 1;
            } else if (nums[mid] < target) {
                low = mid + 1;
            } else {
                return mid;
            }
        }

        return low;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 13:28

Runtime

0 ms | Beats 100.00%

Memory

42.94 MB | Beats 41.40%

Sorry, there are not enough accepted submissions to show data

Code | Java

```java
class Solution {
    public int searchInsert(int[] nums, int target) {
        return binarySearch(nums, target);
```

```java
class Solution {
    public int searchInsert(int[] nums, int target) {
        return binarySearch(nums, target);
    }

    private int binarySearch(int[] nums, int target) {
        int low = 0;
        int high = nums.length - 1;

        while (low <= high) {
            int mid = (low + high) / 2;

            if (nums[mid] > target) {
                high = mid - 1;
            } else if (nums[mid] < target) {
                low = mid + 1;
            } else {
                return mid;
            }
        }

        return low;
```

Ln 11, Col 40 | Saved

Judging... Speed Up

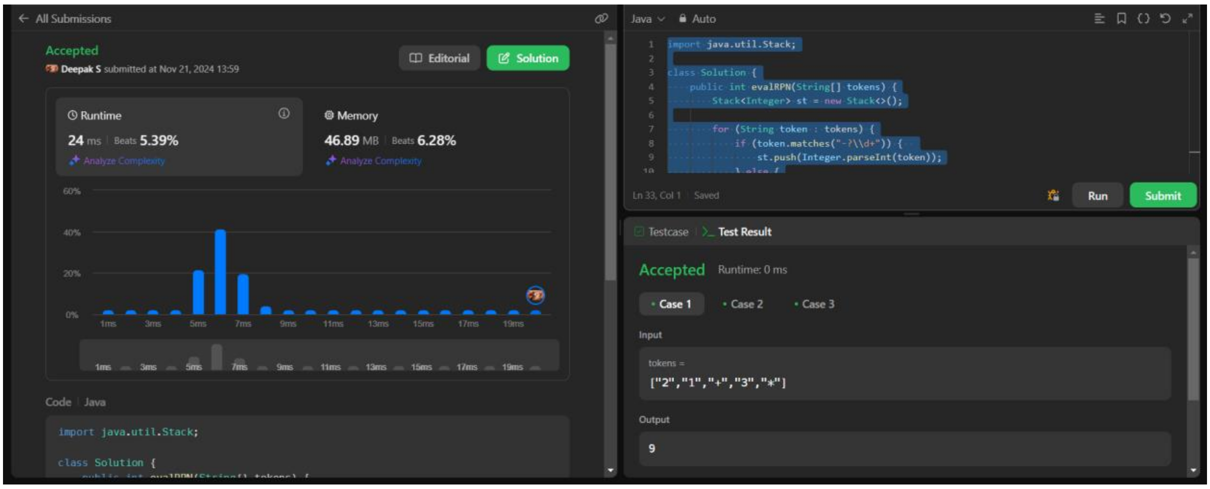
Accepted Runtime: 0 ms

• Case 1 • Case 2 • Case 3

10.Evaluate Reverse Polish Notation

```java
import java.util.Stack;

class Solution {
    public int evalRPN(String[] tokens) {
        Stack<Integer> st = new Stack<>();

        for (String token : tokens) {
            if (token.matches("-?\\d+")) {
                st.push(Integer.parseInt(token));
            } else {
                int b = st.pop();
                int a = st.pop();
                switch (token) {
                    case "+":
                        st.push(a + b);
                        break;
                    case "-":
                        st.push(a - b);
                        break;
                    case "*":
                        st.push(a * b);
                        break;
                    case "/":
                        st.push(a / b);
                        break;
                }
            }
        }

        return st.peek();
    }
}
```

Output:



Accepted
Deepak S submitted at Nov 21, 2024 13:59

Runtime
24 ms | Beats 5.39%

Memory
46.89 MB | Beats 6.28%

Code | Java

```java
import java.util.Stack;

class Solution {
```

```java
import java.util.Stack;

class Solution {
    public int evalRPN(String[] tokens) {
        Stack<Integer> st = new Stack<>();

        for (String token : tokens) {
            if (token.matches("-?\\d+")) {
                st.push(Integer.parseInt(token));
            } else {
```

Ln 33, Col 1 | Saved

Accepted Runtime: 0 ms

Input

tokens =
["2","1","+","3","*"]

Output

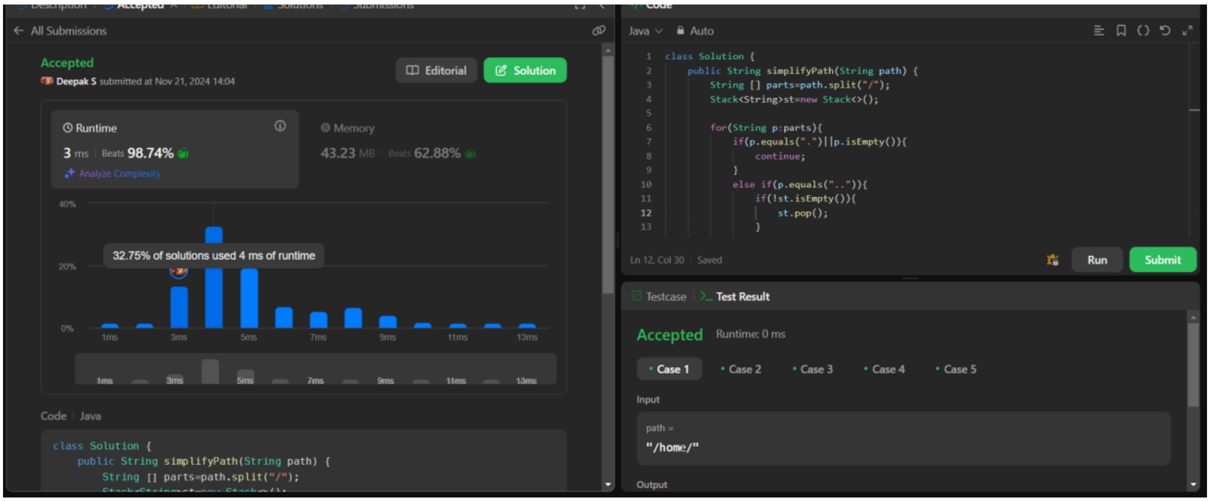
9

11.Simplify Path

Code

```
class Solution {
    public String simplifyPath(String path) {
        String [] parts=path.split("/");
        Stack<String>st=new Stack<>();

        for(String p:parts){
            if(p.equals(".")||p.isEmpty()){
                continue;
            }
            else if(p.equals("..")){
                if(!st.isEmpty()){
                    st.pop();
                }

            }
            else{
                st.push(p);
            }
        }
        StringBuilder res=new StringBuilder();
        for(String p:st){
            res.append('/').append(p);
        }

        return res.length()>0?res.toString():"/";

    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 14:04

Runtime
3 ms | Beats 98.74%

Memory
43.23 MB | Beats 62.88%

32.75% of solutions used 4 ms of runtime

```
class Solution {
    public String simplifyPath(String path) {
        String [] parts=path.split("/");
        Stack<String>st=new Stack<>();

        for(String p:parts){
            if(p.equals(".")||p.isEmpty()){
                continue;
            }
            else if(p.equals("..")){
                if(!st.isEmpty()){
                    st.pop();
                }
```

Accepted Runtime: 0 ms

Input

path =
"/home/"

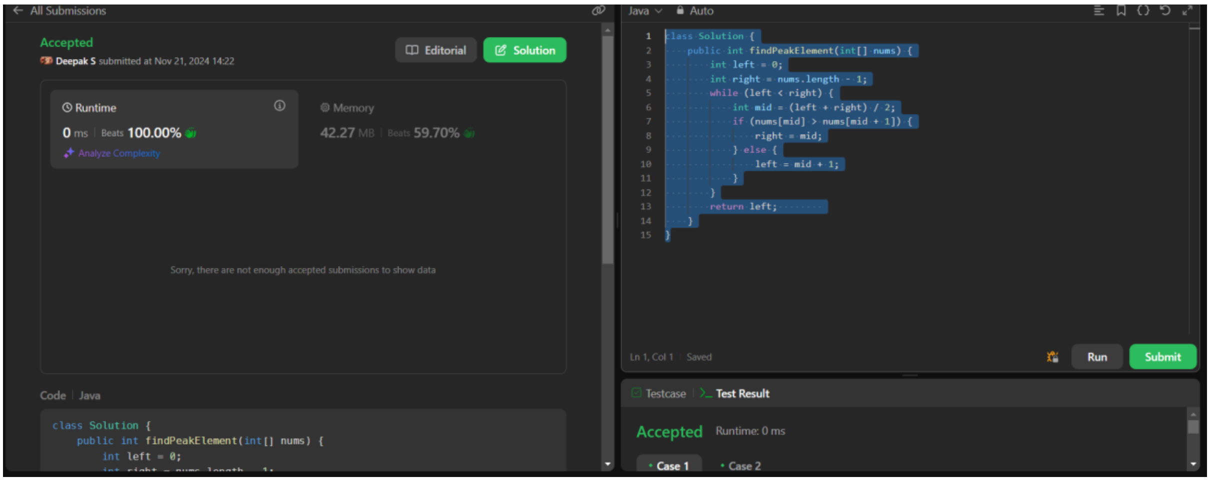
Output

12.Find Peak Element

Code:

```java
class Solution {
    public int findPeakElement(int[] nums) {
        int left = 0;
        int right = nums.length - 1;
        while (left < right) {
            int mid = (left + right) / 2;
            if (nums[mid] > nums[mid + 1]) {
                right = mid;
            } else {
                left = mid + 1;
            }
        }
        return left;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 21, 2024 14:22

Runtime

0 ms | Beats 100.00%

Memory

42.27 MB | Beats 59.70%

Sorry, there are not enough accepted submissions to show data

Code | Java

```java
class Solution {
    public int findPeakElement(int[] nums) {
        int left = 0;
```

```java
class Solution {
    public int findPeakElement(int[] nums) {
        int left = 0;
        int right = nums.length - 1;
        while (left < right) {
            int mid = (left + right) / 2;
            if (nums[mid] > nums[mid + 1]) {
                right = mid;
            } else {
                left = mid + 1;
            }
        }
        return left;
    }
}
```

Ln 1, Col 1 | Saved

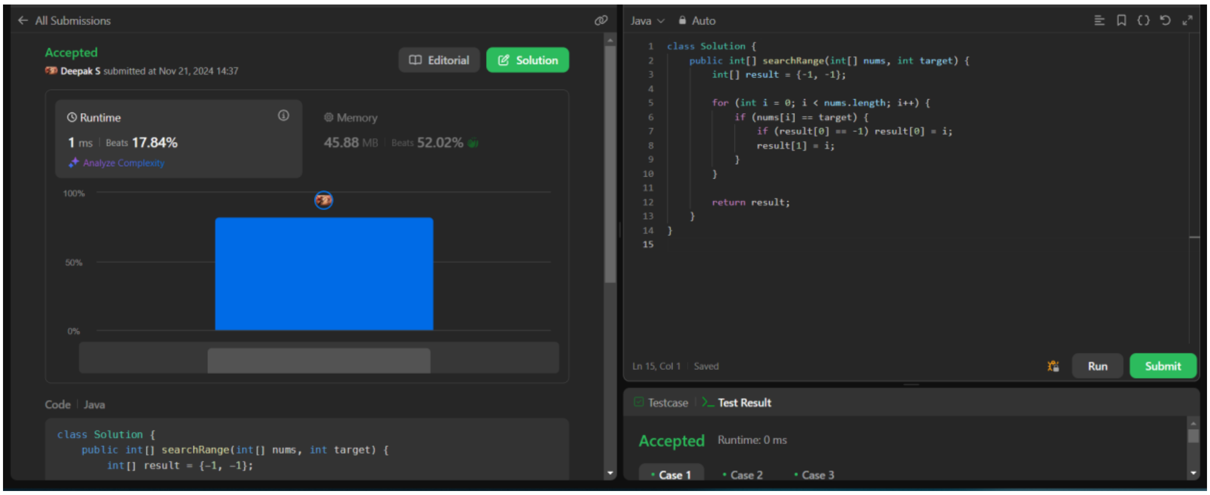
Accepted Runtime: 0 ms

13.Find First and Last Position of Element in Sorted Array

Code:

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == target) {
                if (result[0] == -1) result[0] = i;
                result[1] = i;
            }
        }

        return result;
    }
}
```

Output:



Accepted
Deepak S submitted at Nov 21, 2024 14:37

Runtime
1 ms | Beats 17.84%
Analyze Complexity

Memory
45.88 MB | Beats 52.02%

Code | Java

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};
```

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == target) {
                if (result[0] == -1) result[0] = i;
                result[1] = i;
            }
        }

        return result;
    }
}
```

Ln 15, Col 1 | Saved

Accepted Runtime: 0 ms

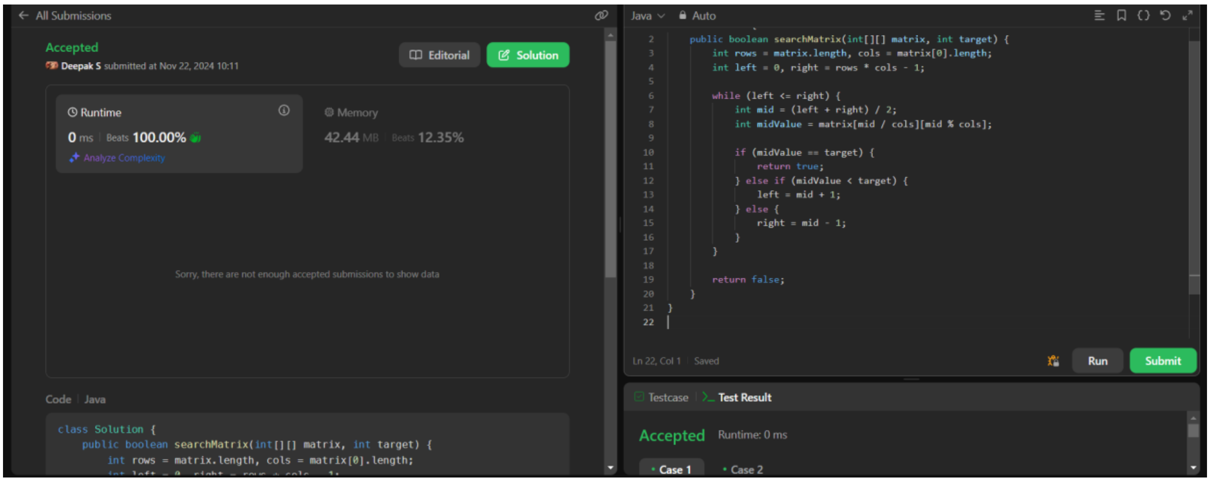
Case 1 Case 2 Case 3

14.Search a 2D Matrix:

Code:

```java
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        int rows = matrix.length, cols = matrix[0].length;
        int left = 0, right = rows * cols - 1;

        while (left <= right) {
            int mid = (left + right) / 2;
            int midValue = matrix[mid / cols][mid % cols];

            if (midValue == target) {
                return true;
            } else if (midValue < target) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }

        return false;
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 22, 2024 10:11

Runtime

0 ms | Beats 100.00%

Memory

42.44 MB | Beats 12.35%

Sorry, there are not enough accepted submissions to show data

Code | Java

```java
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        int rows = matrix.length, cols = matrix[0].length;
        int left = 0, right = rows * cols - 1;
```

```java
    public boolean searchMatrix(int[][] matrix, int target) {
        int rows = matrix.length, cols = matrix[0].length;
        int left = 0, right = rows * cols - 1;

        while (left <= right) {
            int mid = (left + right) / 2;
            int midValue = matrix[mid / cols][mid % cols];

            if (midValue == target) {
                return true;
            } else if (midValue < target) {
                left = mid + 1;
            } else {
                right = mid - 1;
            }
        }

        return false;
    }
}
```

Ln 22, Col 1 | Saved

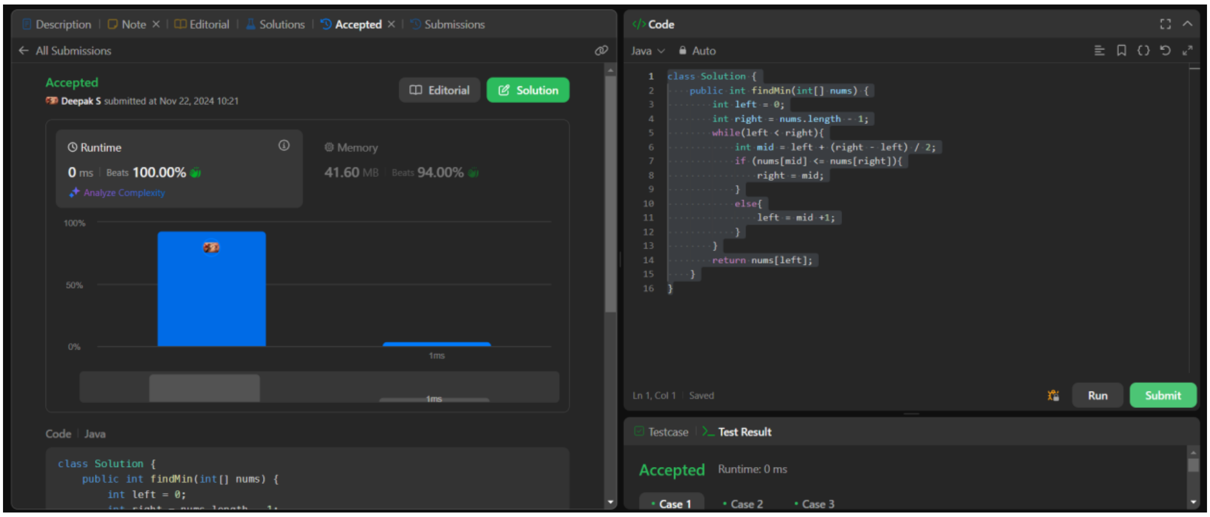
Accepted Runtime: 0 ms

15.Find Minimum in Rotated Sorted Array

Code:

```
class Solution {
    public int findMin(int[] nums) {
        int left = 0;
        int right = nums.length - 1;
        while(left < right){
            int mid = left + (right - left) / 2;
            if (nums[mid] <= nums[right]){
                right = mid;
            }
            else{
                left = mid +1;
            }
        }
        return nums[left];
    }
}
```

Output:



Accepted

Deepak S submitted at Nov 22, 2024 10:21

0 ms | Beats 100.00%

41.60 MB | Beats 94.00%

Code | Java

```
class Solution {
    public int findMin(int[] nums) {
        int left = 0;
```

```
class Solution {
    public int findMin(int[] nums) {
        int left = 0;
        int right = nums.length - 1;
        while(left < right){
            int mid = left + (right - left) / 2;
            if (nums[mid] <= nums[right]){
                right = mid;
            }
            else{
                left = mid +1;
            }
        }
        return nums[left];
    }
}
```

Ln 1, Col 1 | Saved

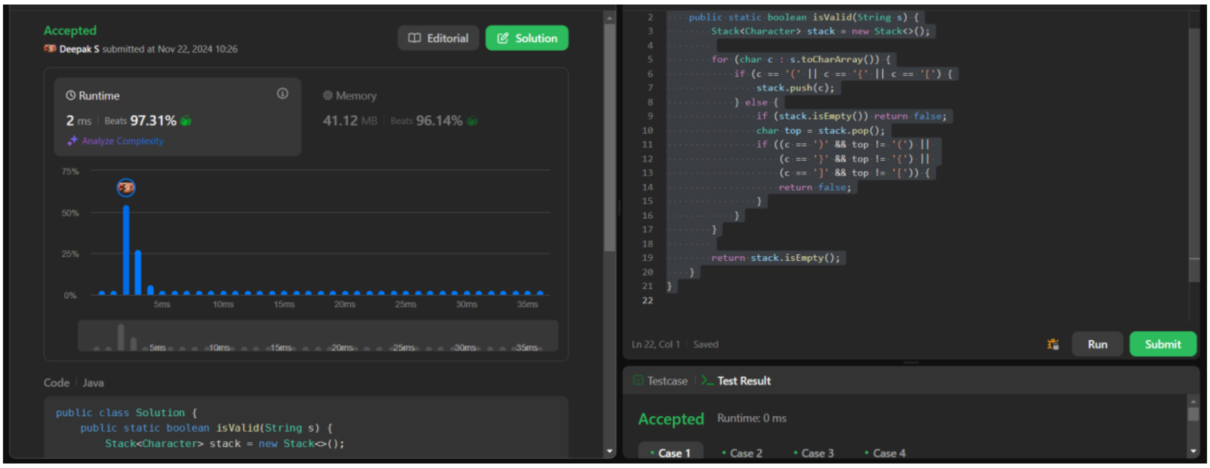
Accepted Runtime: 0 ms

16.Valid Parentheses

Code:

```java
public class Solution {
    public static boolean isValid(String s) {
        Stack<Character> stack = new Stack<>();

        for (char c : s.toCharArray()) {
            if (c == '(' || c == '{' || c == '[') {
                stack.push(c);
            } else {
                if (stack.isEmpty()) return false;
                char top = stack.pop();
                if ((c == ')' && top != '(') ||
                    (c == '}' && top != '{') ||
                    (c == ']' && top != '[')) {
                    return false;
                }
            }
        }

        return stack.isEmpty();
    }
}
```

Output:



Accepted
Deepak S submitted at Nov 22, 2024 10:26

Runtime
2 ms | Beats 97.31%

Memory
41.12 MB | Beats 96.14%

Code | Java

```java
public class Solution {
    public static boolean isValid(String s) {
        Stack<Character> stack = new Stack<>();
```

```java
    public static boolean isValid(String s) {
        Stack<Character> stack = new Stack<>();

        for (char c : s.toCharArray()) {
            if (c == '(' || c == '{' || c == '[') {
                stack.push(c);
            } else {
                if (stack.isEmpty()) return false;
                char top = stack.pop();
                if ((c == ')' && top != '(') ||
                    (c == '}' && top != '{') ||
                    (c == ']' && top != '[')) {
                    return false;
                }
            }
        }

        return stack.isEmpty();
    }
}
```

Ln 22, Col 1 | Saved

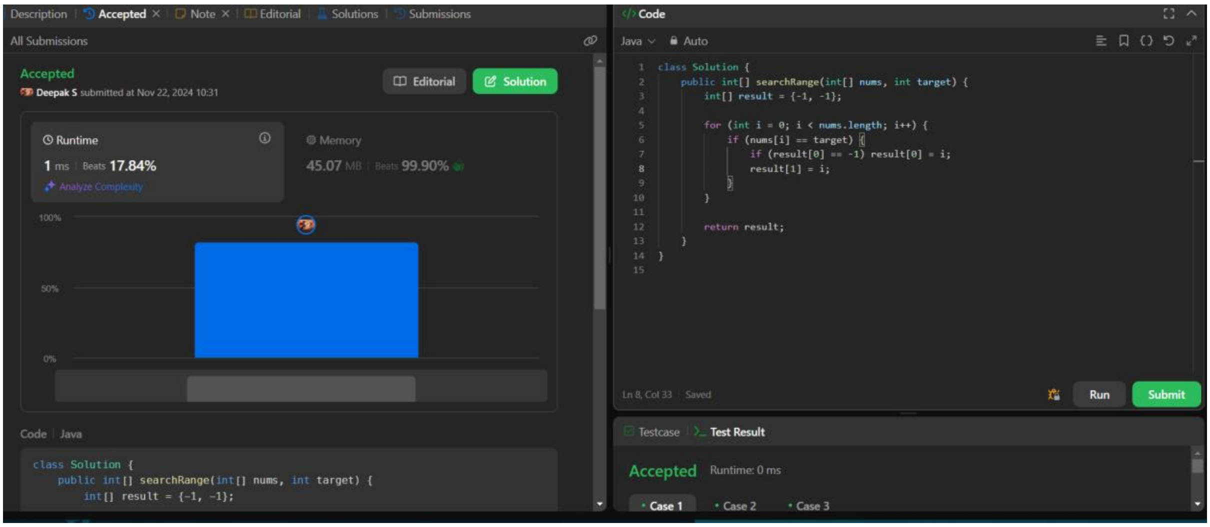
Accepted Runtime: 0 ms

17.Find First and Last Position of Element in Sorted Array

Code:

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == target) {
                if (result[0] == -1) result[0] = i;
                result[1] = i;
            }
        }

        return result;
    }
}
```

Output:



Accepted
Deepak S submitted at Nov 22, 2024 10:31

Runtime
1 ms | Beats 17.84%
Analyze Complexity

Memory
45.07 MB | Beats 99.90%

Code | Java

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};
```

```java
class Solution {
    public int[] searchRange(int[] nums, int target) {
        int[] result = {-1, -1};

        for (int i = 0; i < nums.length; i++) {
            if (nums[i] == target) {
                if (result[0] == -1) result[0] = i;
                result[1] = i;
            }
        }

        return result;
    }
}
```

Ln 8, Col 33 | Saved

Accepted Runtime: 0 ms