**TASK REPORT**

**ALGORITHMS AND DATA STRUCTURE**

**Search Algorithms**

**Created by:**

**Onic Agustino**

**L200234275**

**X**

**Date : April 29, 2025**

**INFORMATICS ENGINEERING**

**FACULTY OF COMMUNICATION AND INFORMATICS**

**UNIVERSITAS MUHAMMADIYAH SURAKARTA**

**2024/2025**

**Leetcode**

1. **Problem 1: Kth Missing Positive Number**

* **Problem Explanation**

You are given a sorted array arr of positive integers and an integer k.  
You need to find the k-th positive integer that is missing from the array.

**Input:**

* arr (list of positive integers, sorted)
* k (positive integer)

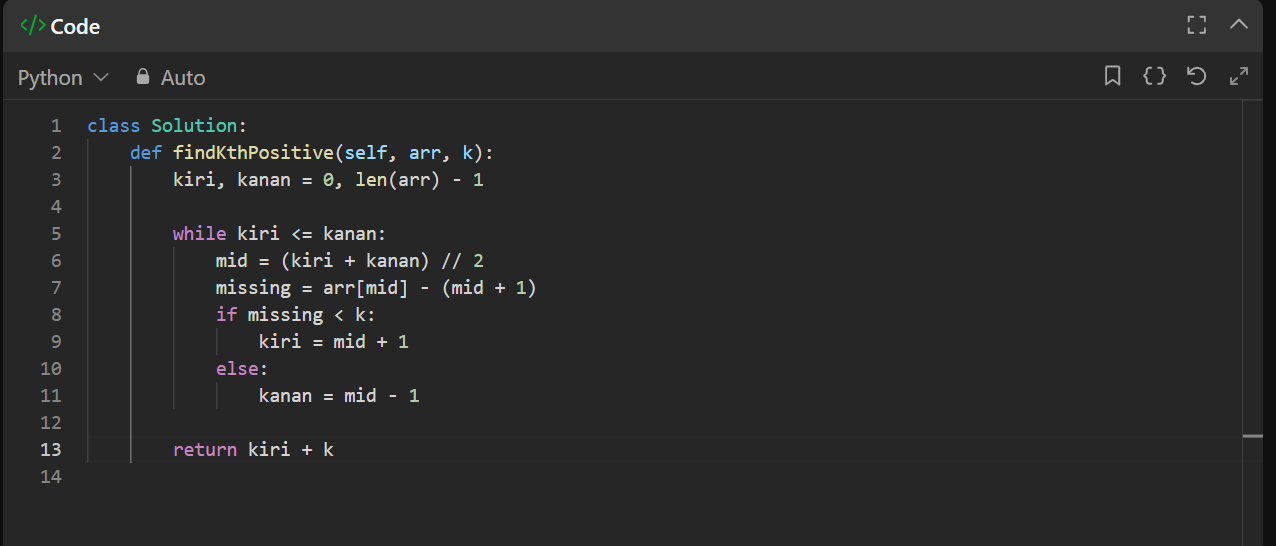
**` Output:**

* Return the k-th missing positive number.
* **Solution Approach**

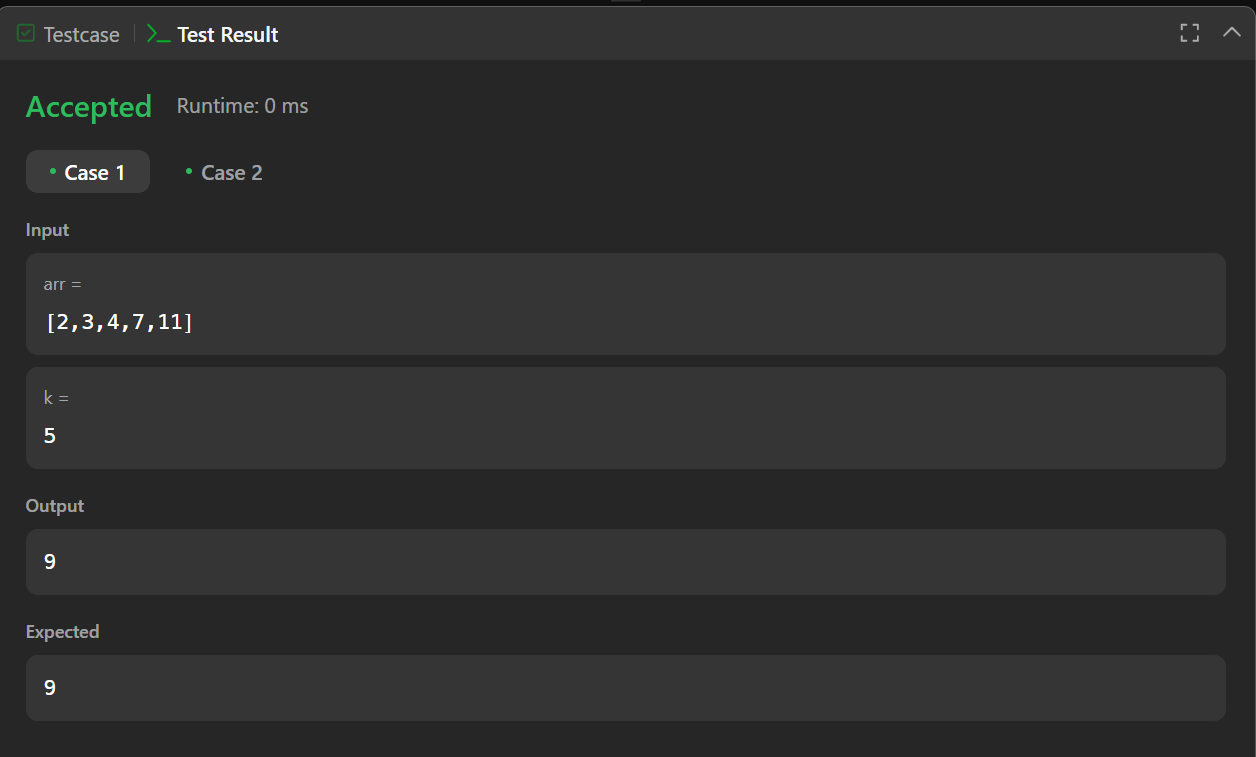
We use **binary search** to optimize the search.  
For each arr[mid], the number of missing numbers before it is arr[mid] - (mid + 1).  
We perform binary search to locate the smallest index where the number of missing integers is at least k.

If not found directly, the final answer will be left + k.

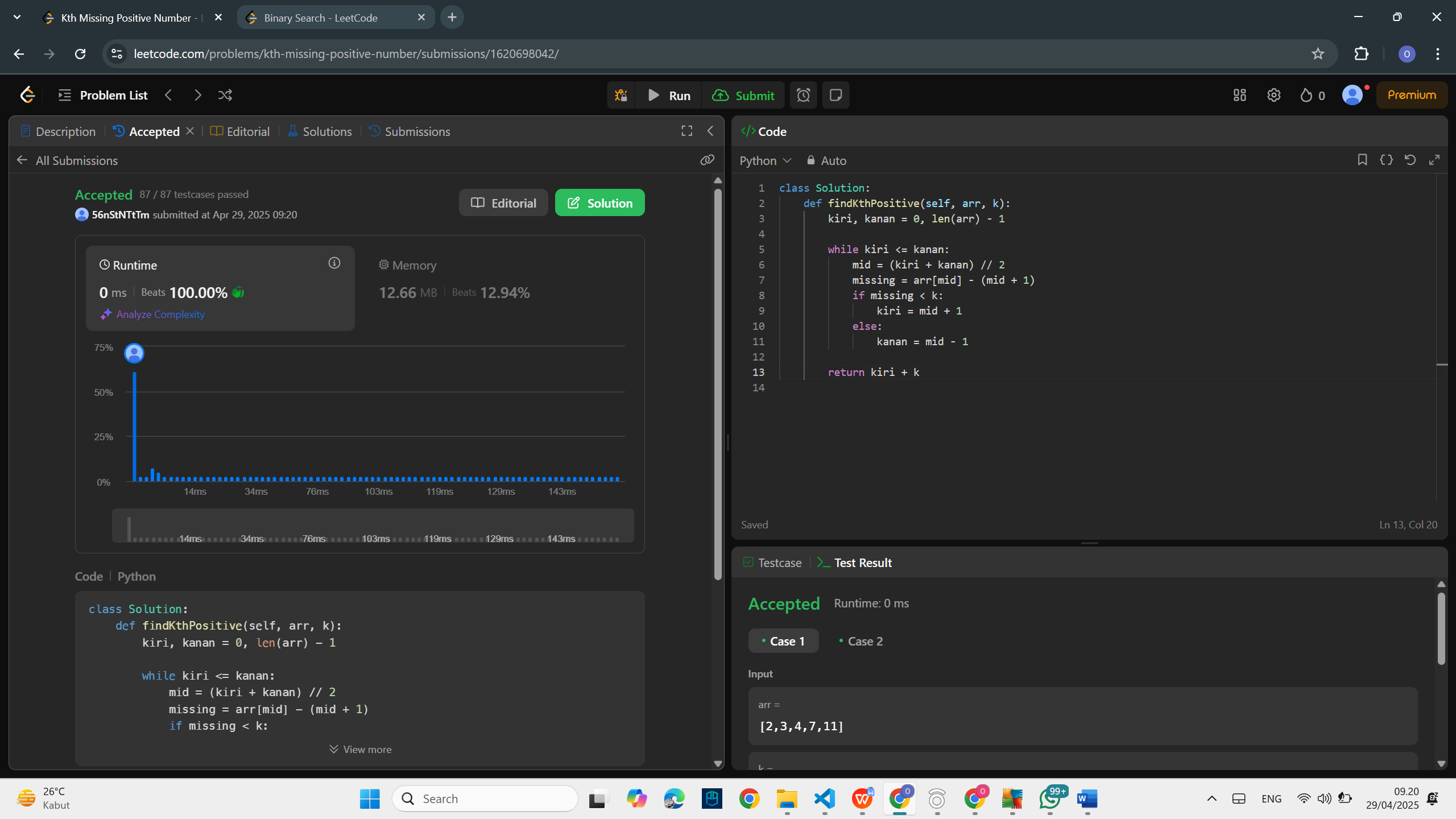
* **Code Implementation**

****

* **Output**

****

* **Screenshot Submission**

****

\

1. **Problem 2: Binary Search**

* **Problem Explanation**

You are given a sorted array nums and a target value.  
You must return the index where the target value is found. If not found, return -1.

**Input:**

* nums (list of integers, sorted in ascending order)
* target (integer)

**Output:**

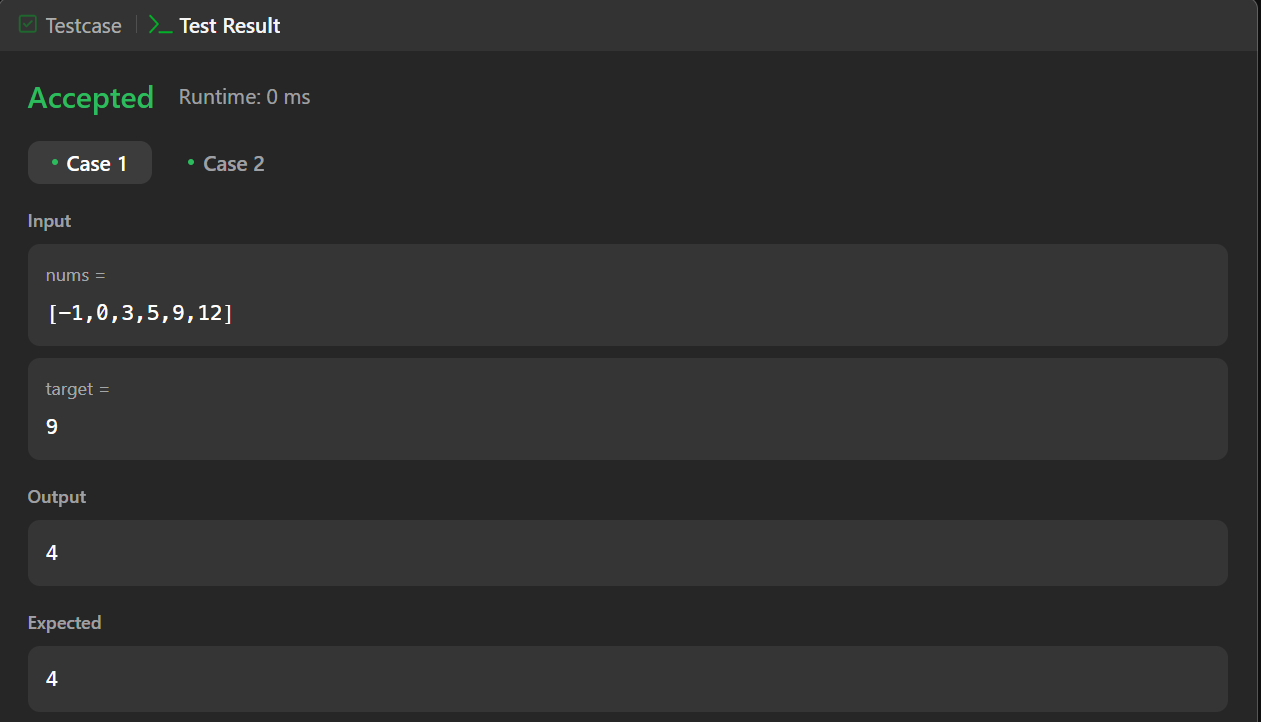
* Index of the target if found, else -1.
* **Solution Approach**

We use **binary search** which has a time complexity of **O(log n)**.  
At each step, compare the target with the middle element and adjust the search boundaries (left and right) accordingly.

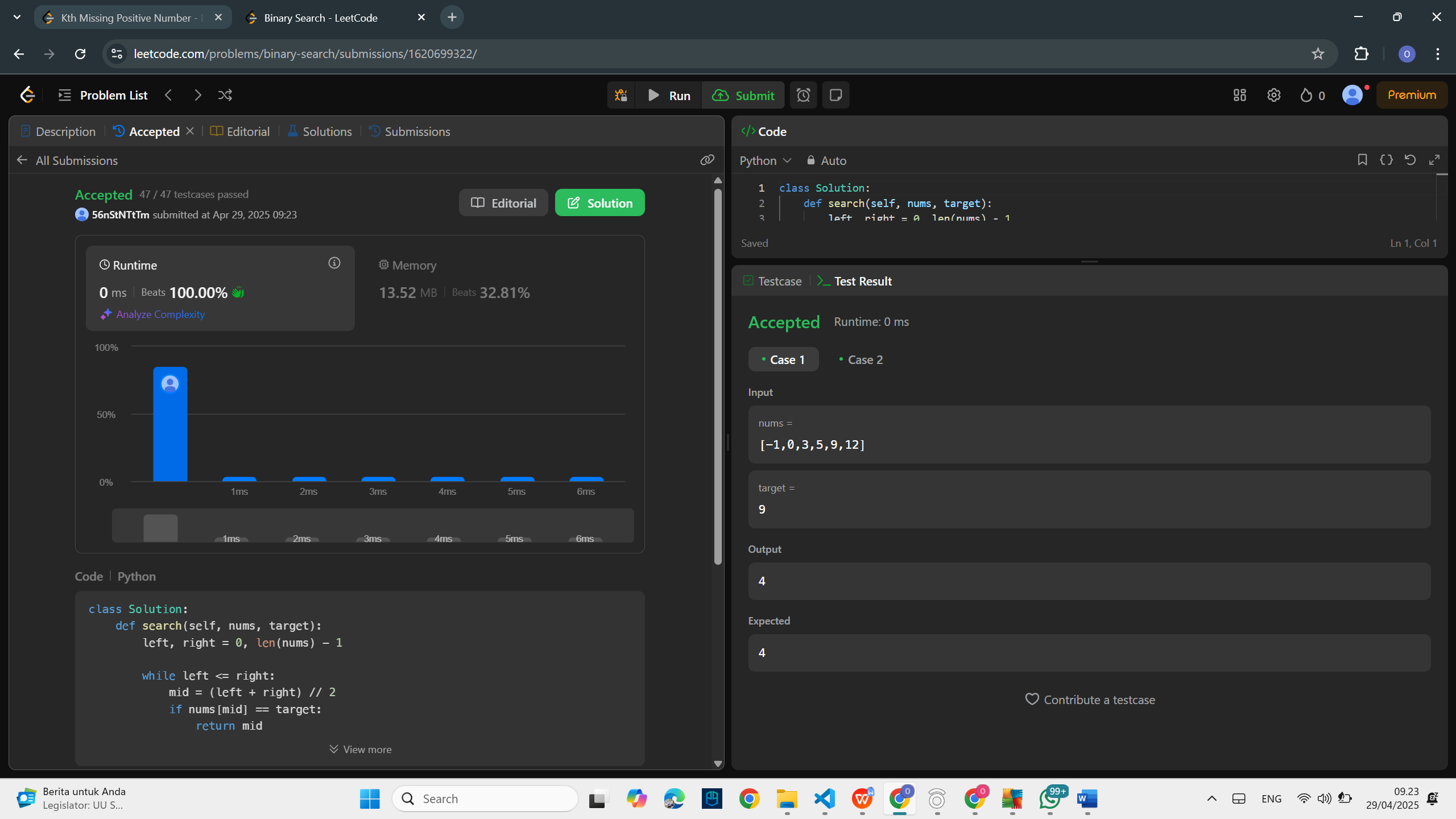
* **Code Implementation**



* **Output**



* **Screenshot Submission**

****