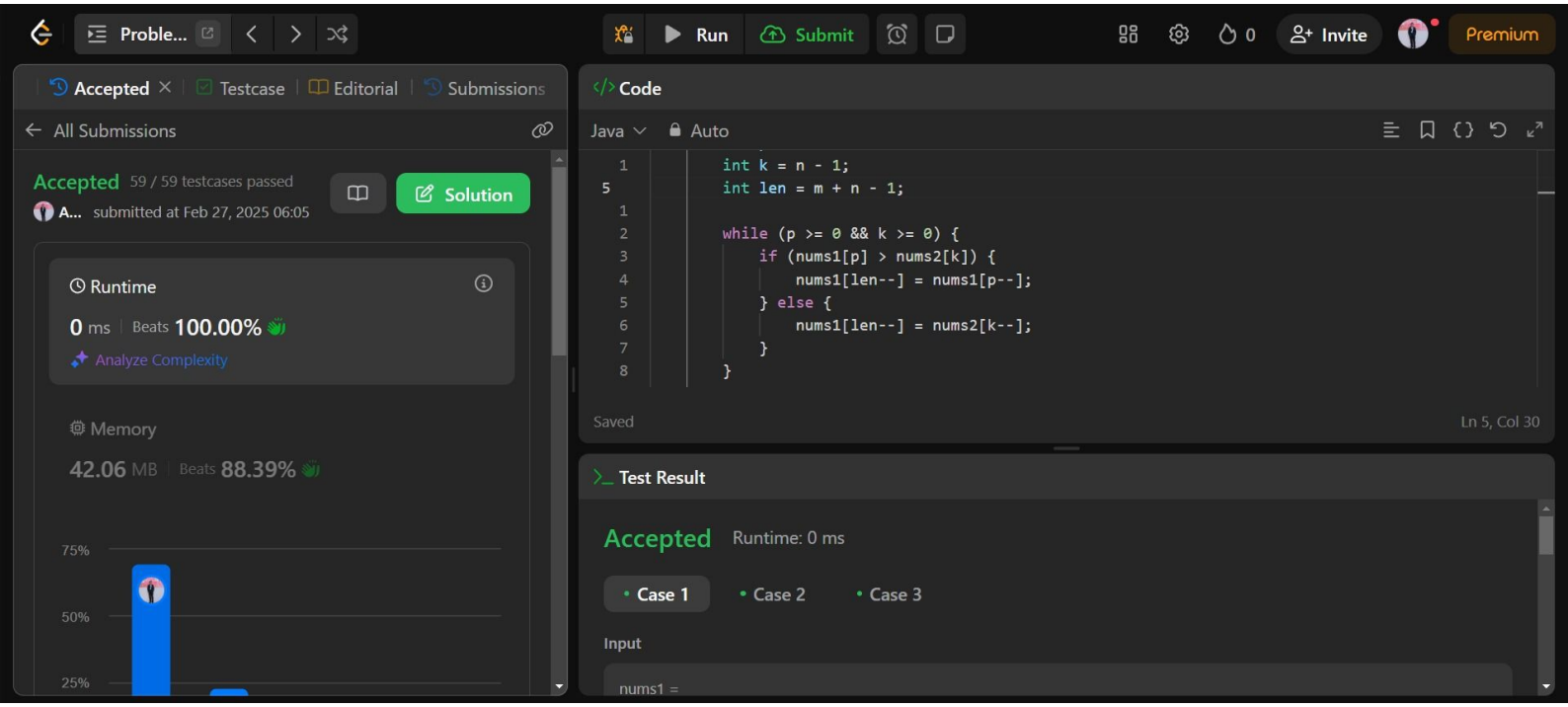
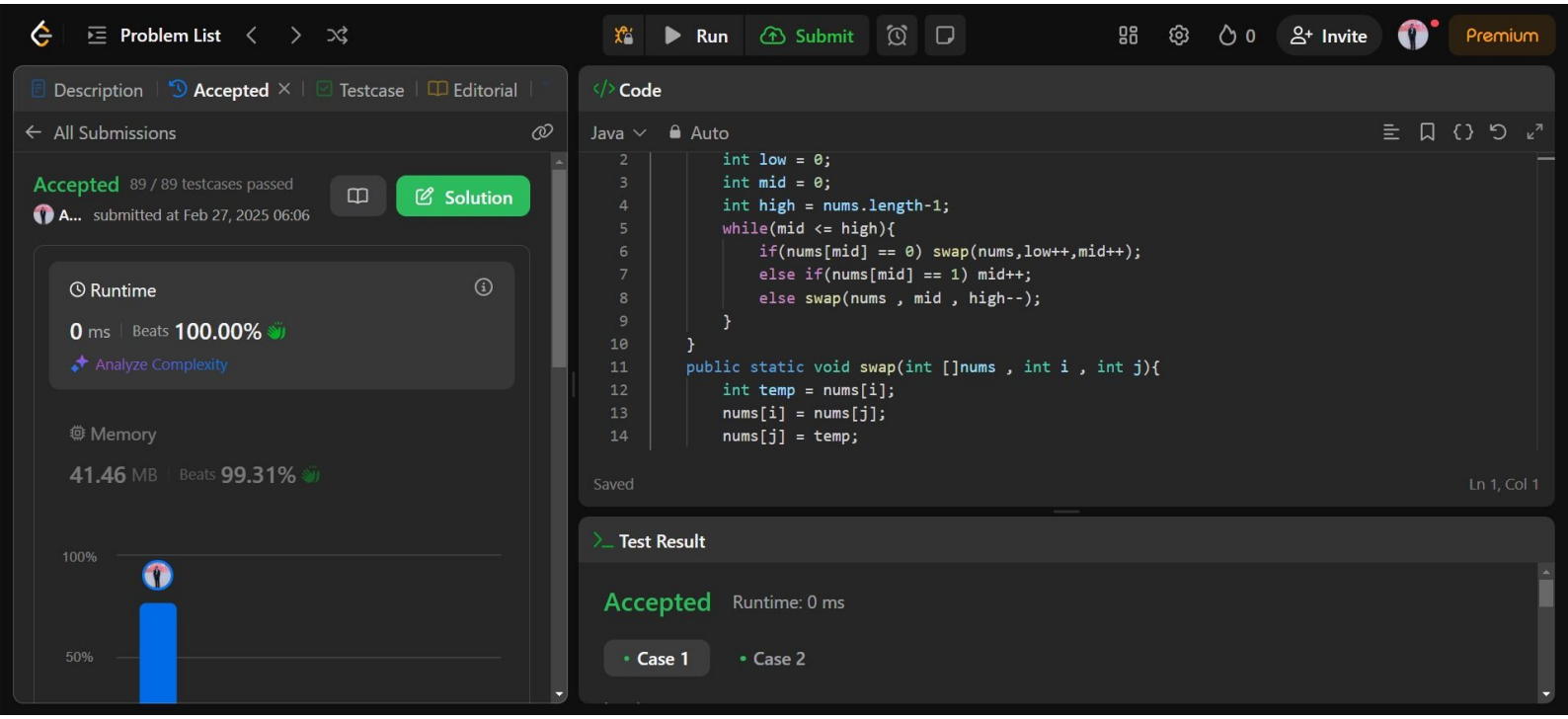
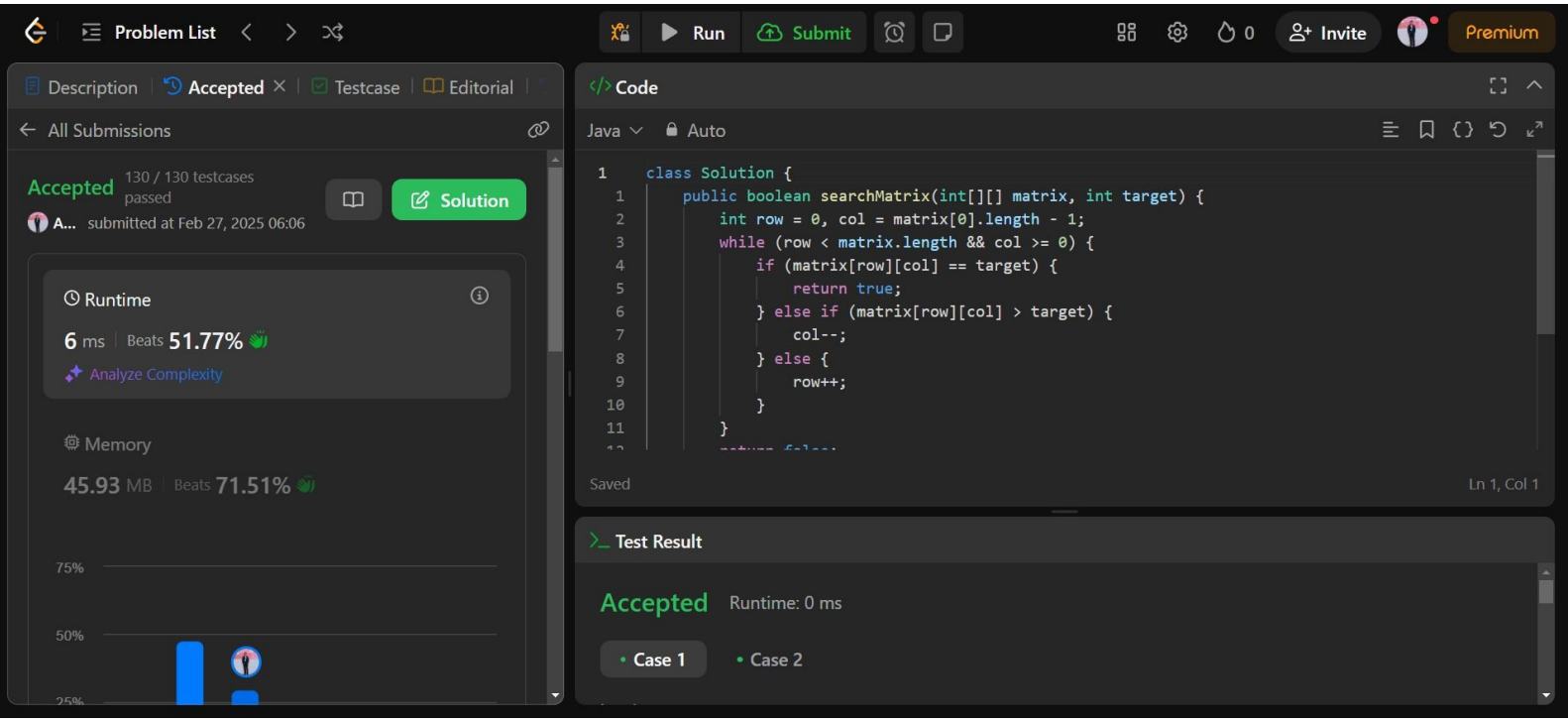
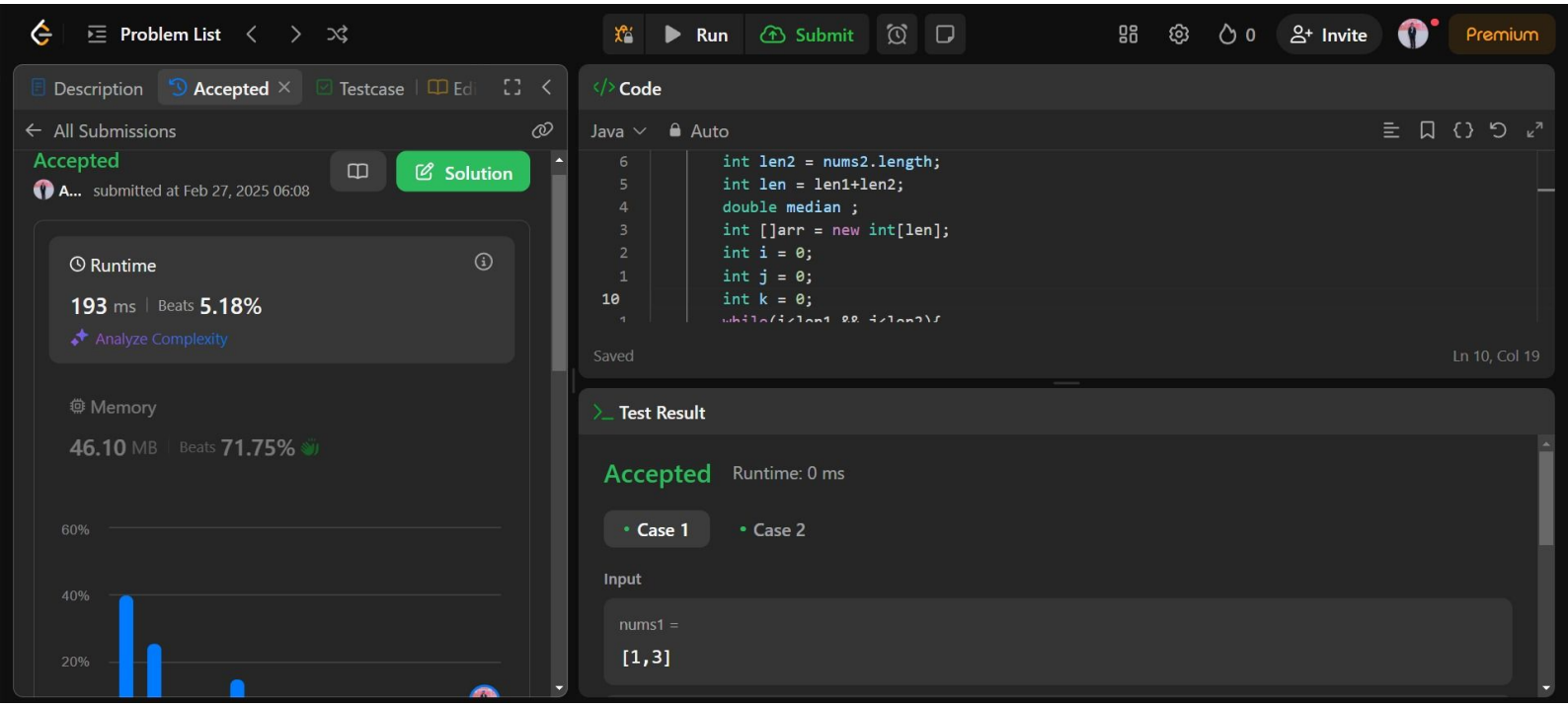
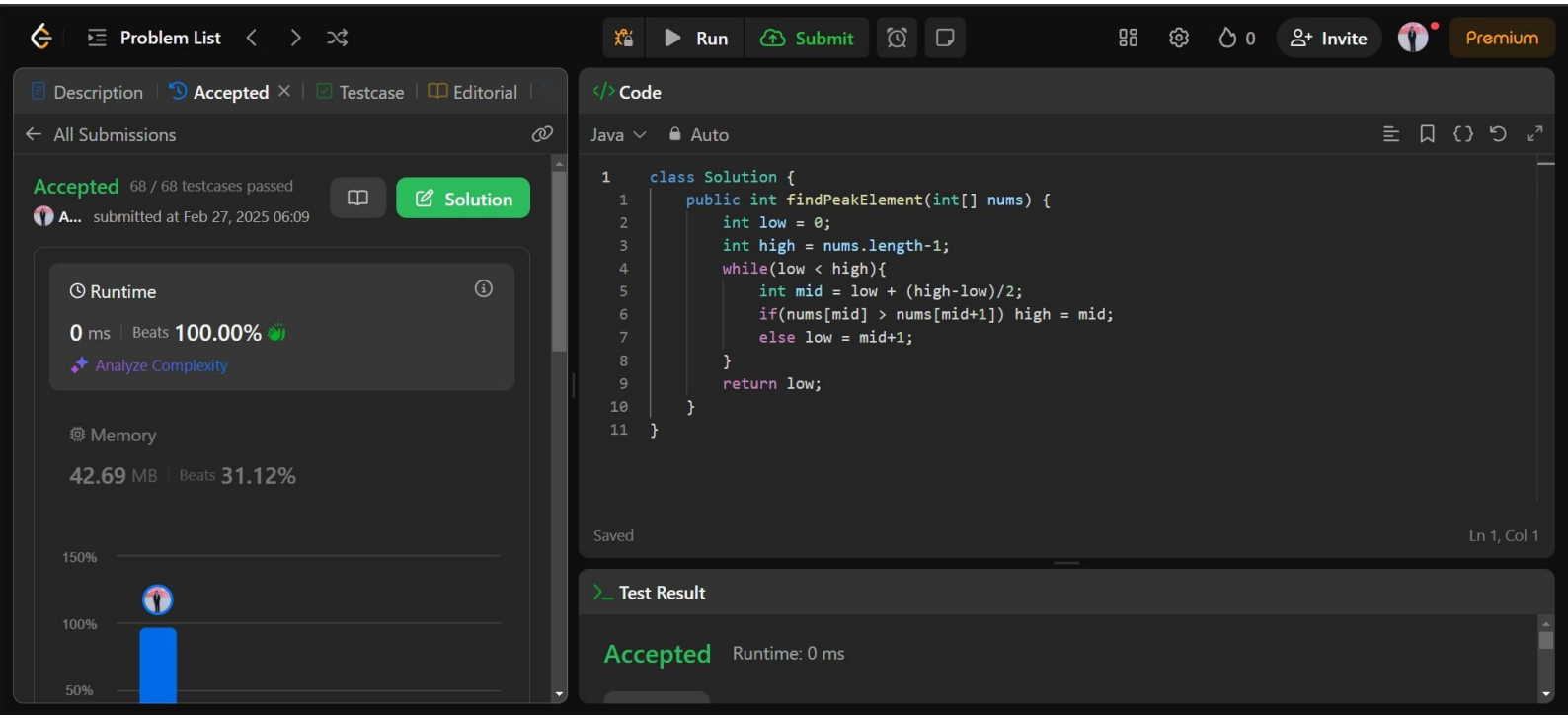

Proble...
Run
Submit
0
Invite
Premium
Accepted
Testcase
Editorial
Submissions
All Submissions
Accepted 59 / 59 testcases passed
A... submitted at Feb 27, 2025 06:05
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
42.06 MB | Beats 88.39%
75%
50%
25%
Code
Java
Auto
1 int k = n - 1;
5 int len = m + n - 1;
1
2 while (p >= 0 && k >= 0) {
3 if (nums1[p] > nums2[k]) {
4 nums1[len--] = nums1[p--];
5 } else {
6 nums1[len--] = nums2[k--];
7 }
8 }
Saved
Ln 5, Col 30
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Case 3
Input
nums1 =


Problem List
Run
Submit
Invite
Premium
Description
Accepted
Testcase
Editorial
All Submissions
Accepted 89 / 89 testcases passed
A... submitted at Feb 27, 2025 06:06
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
41.46 MB | Beats 99.31%
100%
50%
Code
Java
Auto
int low = 0;
int mid = 0;
int high = nums.length-1;
while(mid <= high){
if(nums[mid] == 0) swap(nums,low++,mid++);
else if(nums[mid] == 1) mid++;
else swap(nums , mid , high--);
}
}
public static void swap(int []nums , int i , int j){
int temp = nums[i];
nums[i] = nums[j];
nums[j] = temp;
Saved
Ln 1, Col 1
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2


Problem List
Run
Submit
Invite
Premium
Description
Accepted
Testcase
Editorial
All Submissions
Accepted
130 / 130 testcases passed
A... submitted at Feb 27, 2025 06:06
Solution
Runtime
6 ms | Beats 51.77%
Analyze Complexity
Memory
45.93 MB | Beats 71.51%
75%
50%
25%
Code
Java
Auto
1 class Solution {
1 public boolean searchMatrix(int[][] matrix, int target) {
2 int row = 0, col = matrix[0].length - 1;
3 while (row < matrix.length && col >= 0) {
4 if (matrix[row][col] == target) {
5 return true;
6 } else if (matrix[row][col] > target) {
7 col--;
8 } else {
9 row++;
10 }
11 }
Saved
Ln 1, Col 1
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2


Problem List
Run
Submit
Invite
Premium
Description
Accepted
Testcase
All Submissions
Accepted
submitted at Feb 27, 2025 06:08
Solution
Runtime
193 ms | Beats 5.18%
Analyze Complexity
Memory
46.10 MB | Beats 71.75%
60%
40%
20%
Code
Java
Auto
int len2 = nums2.length;
int len = len1+len2;
double median ;
int []arr = new int[len];
int i = 0;
int j = 0;
int k = 0;
Saved
Ln 10, Col 19
Test Result
Accepted
Runtime: 0 ms
Case 1
Case 2
Input
nums1 =
[1,3]


Problem List
Run
Submit
0
Invite
Premium
Description
Accepted
Testcase
Editorial
All Submissions
Accepted 68 / 68 testcases passed
A... submitted at Feb 27, 2025 06:09
Solution
Runtime
0 ms | Beats 100.00%
Analyze Complexity
Memory
42.69 MB | Beats 31.12%
150%
100%
50%
Code
Java
Auto
class Solution {
public int findPeakElement(int[] nums) {
int low = 0;
int high = nums.length-1;
while(low < high){
int mid = low + (high-low)/2;
if(nums[mid] > nums[mid+1]) high = mid;
else low = mid+1;
}
return low;
}
}
Saved
Ln 1, Col 1
Test Result
Accepted
Runtime: 0 ms