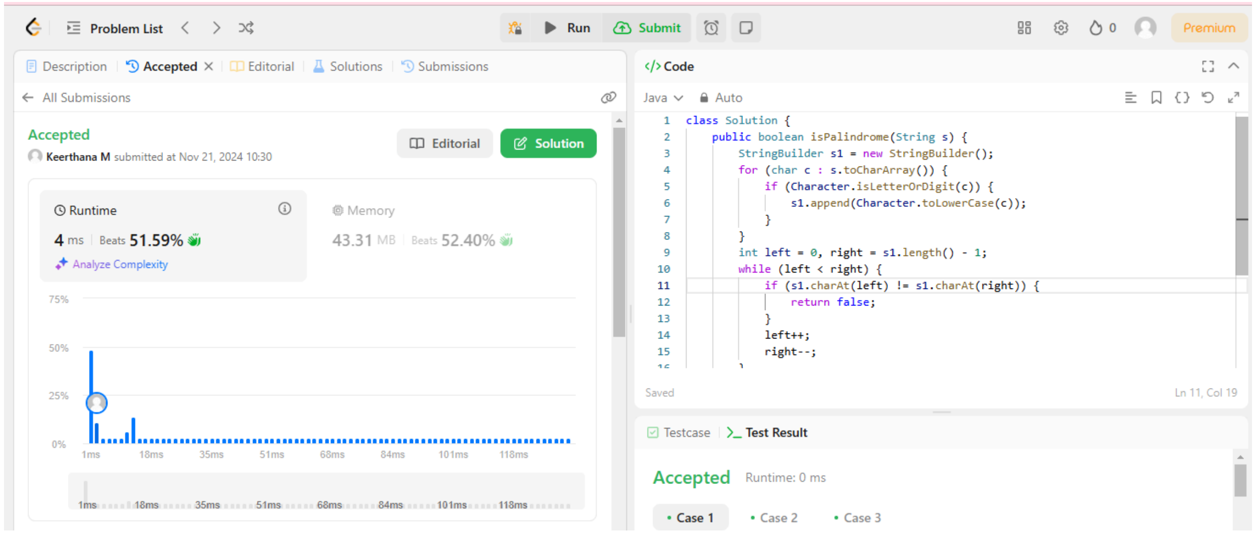
DSA Coding Practice-9

1.Valid Palindrome:



Accepted
Keerthana M submitted at Nov 21, 2024 10:30

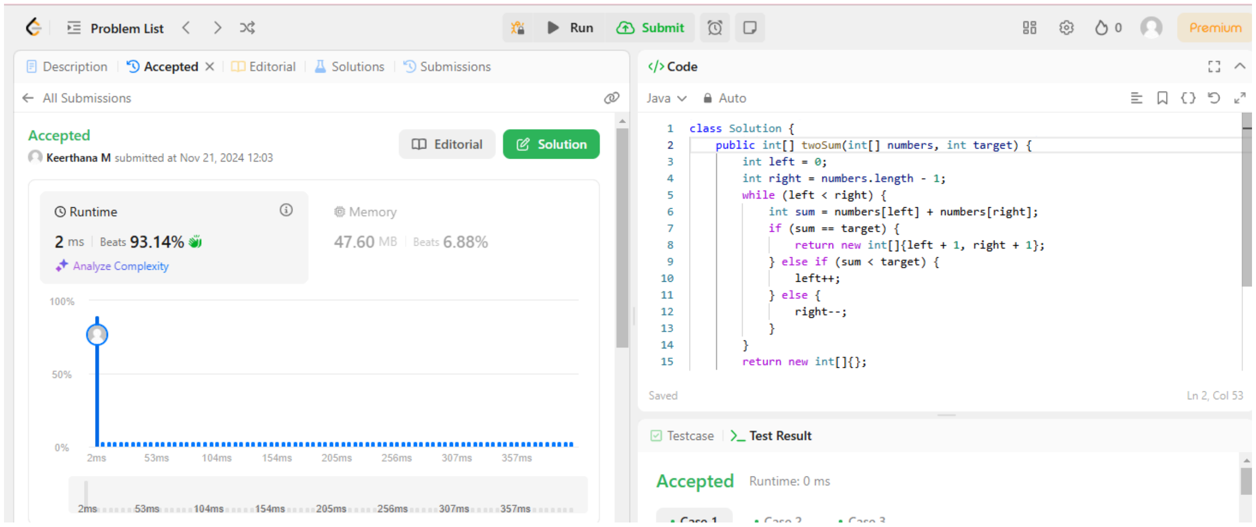
Runtime
4 ms | Beats 51.59%

Memory
43.31 MB | Beats 52.40%

```java
class Solution {
    public boolean isPalindrome(String s) {
        StringBuilder s1 = new StringBuilder();
        for (char c : s.toCharArray()) {
            if (Character.isLetterOrDigit(c)) {
                s1.append(Character.toLowerCase(c));
            }
        }
        int left = 0, right = s1.length() - 1;
        while (left < right) {
            if (s1.charAt(left) != s1.charAt(right)) {
                return false;
            }
            left++;
            right--;
```

Accepted Runtime: 0 ms

2.Two Sum II:



Accepted
Keerthana M submitted at Nov 21, 2024 12:03

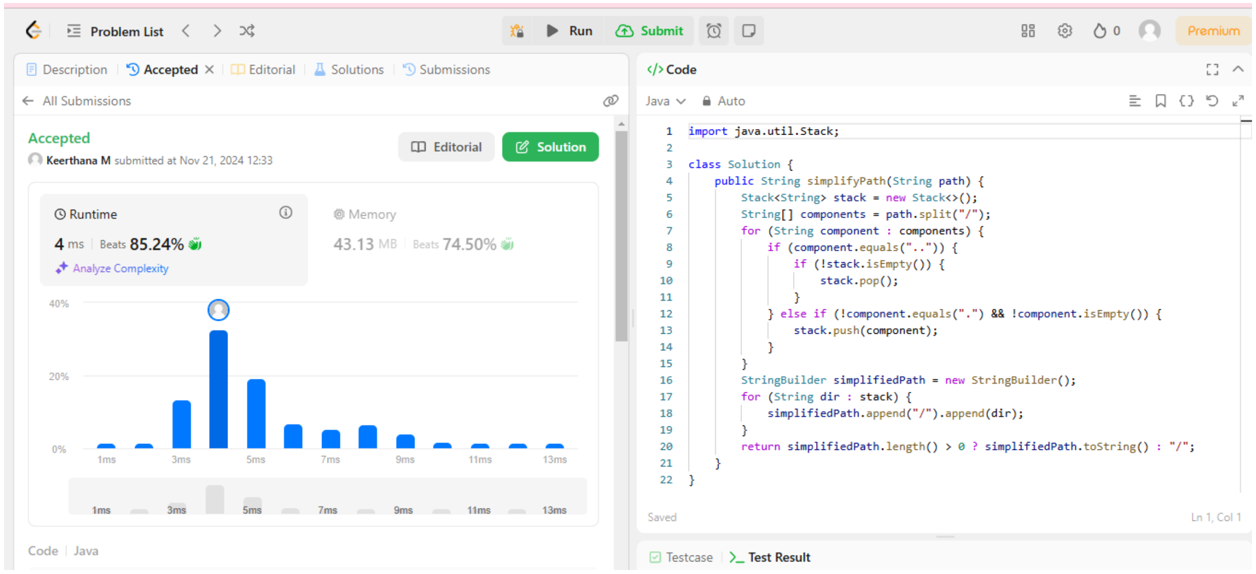
Runtime
2 ms | Beats 93.14%

Memory
47.60 MB | Beats 6.88%

```java
class Solution {
    public int[] twoSum(int[] numbers, int target) {
        int left = 0;
        int right = numbers.length - 1;
        while (left < right) {
            int sum = numbers[left] + numbers[right];
            if (sum == target) {
                return new int[]{left + 1, right + 1};
            } else if (sum < target) {
                left++;
            } else {
                right--;
            }
        }
        return new int[]{};
```

Accepted Runtime: 0 ms

3.Simplify Path:



Accepted
Keerthana M submitted at Nov 21, 2024 12:33

Runtime
4 ms | Beats 85.24%

Memory
43.13 MB | Beats 74.50%

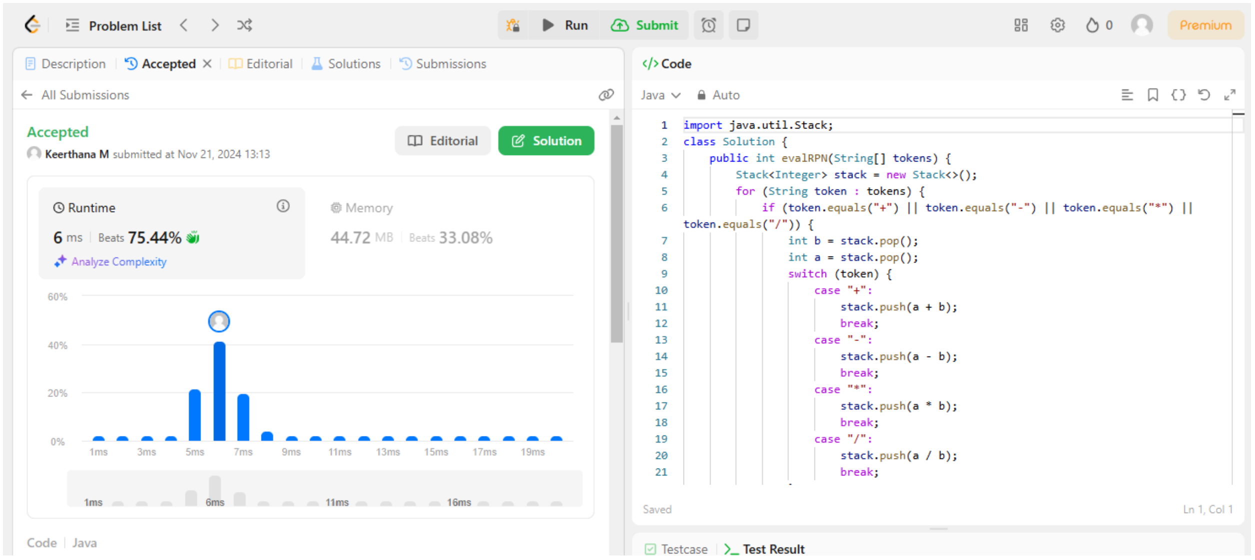
Code | Java

```java
import java.util.Stack;

class Solution {
    public String simplifyPath(String path) {
        Stack<String> stack = new Stack<>();
        String[] components = path.split("/");
        for (String component : components) {
            if (component.equals("..")) {
                if (!stack.isEmpty()) {
                    stack.pop();
                }
            } else if (!component.equals(".") && !component.isEmpty()) {
                stack.push(component);
            }
        }
        StringBuilder simplifiedPath = new StringBuilder();
        for (String dir : stack) {
            simplifiedPath.append("/").append(dir);
        }
        return simplifiedPath.length() > 0 ? simplifiedPath.toString() : "/";
    }
}
```

Saved
Ln 1, Col 1

4.Evaluate Reverse Polish Notation:



Accepted
Keerthana M submitted at Nov 21, 2024 13:13

Runtime
6 ms | Beats 75.44%

Memory
44.72 MB | Beats 33.08%

Code | Java

```java
import java.util.Stack;
class Solution {
    public int evalRPN(String[] tokens) {
        Stack<Integer> stack = new Stack<>();
        for (String token : tokens) {
            if (token.equals("+") || token.equals("-") || token.equals("*") ||
token.equals("/")) {
                int b = stack.pop();
                int a = stack.pop();
                switch (token) {
                    case "+":
                        stack.push(a + b);
                        break;
                    case "-":
                        stack.push(a - b);
                        break;
                    case "*":
                        stack.push(a * b);
                        break;
                    case "/":
                        stack.push(a / b);
                        break;
```

Saved
Ln 1, Col 1