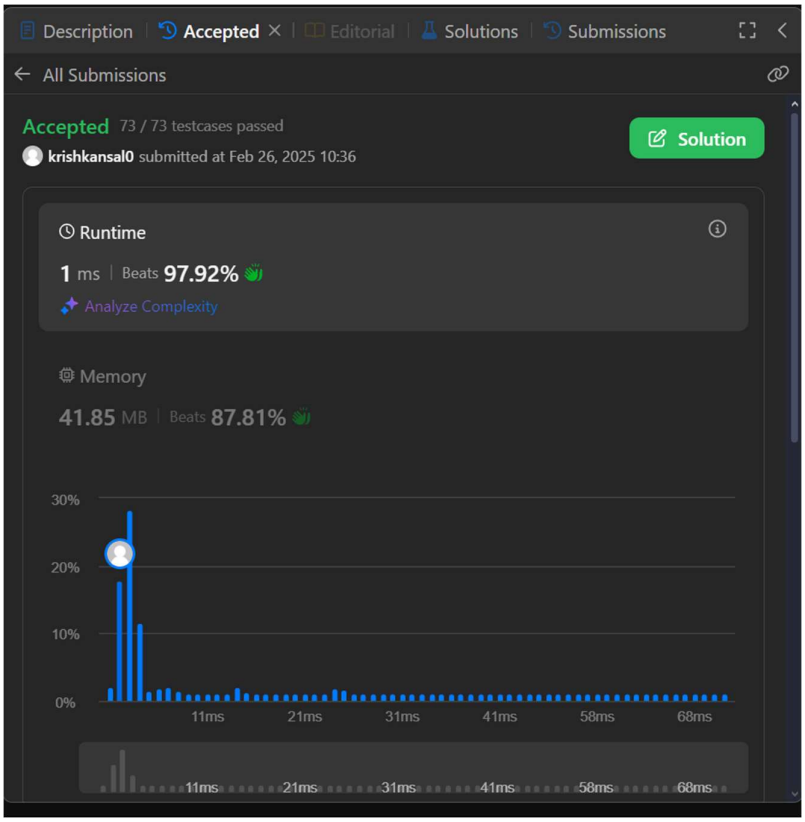
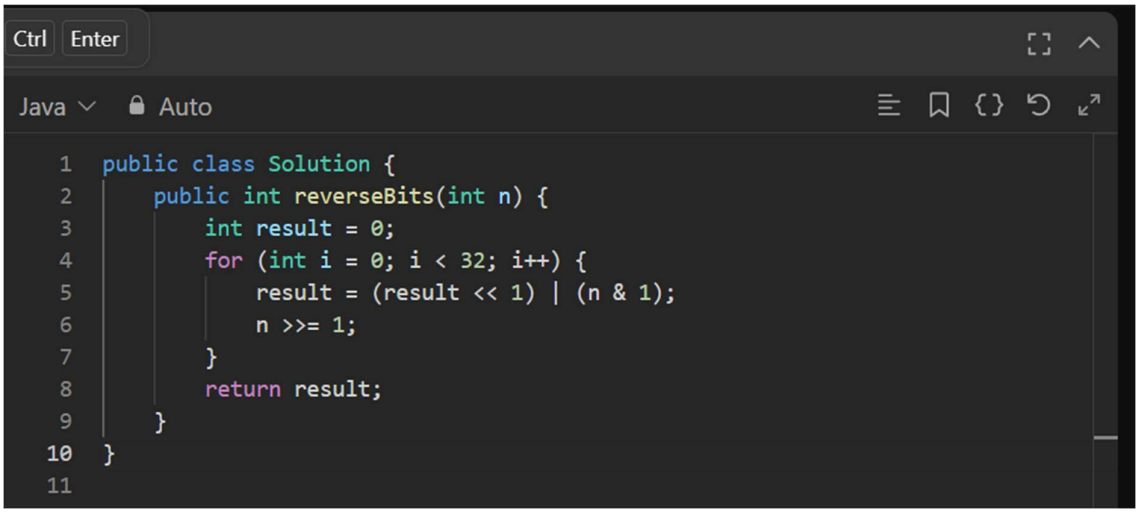
Krish Aggarwal 22BCS16370

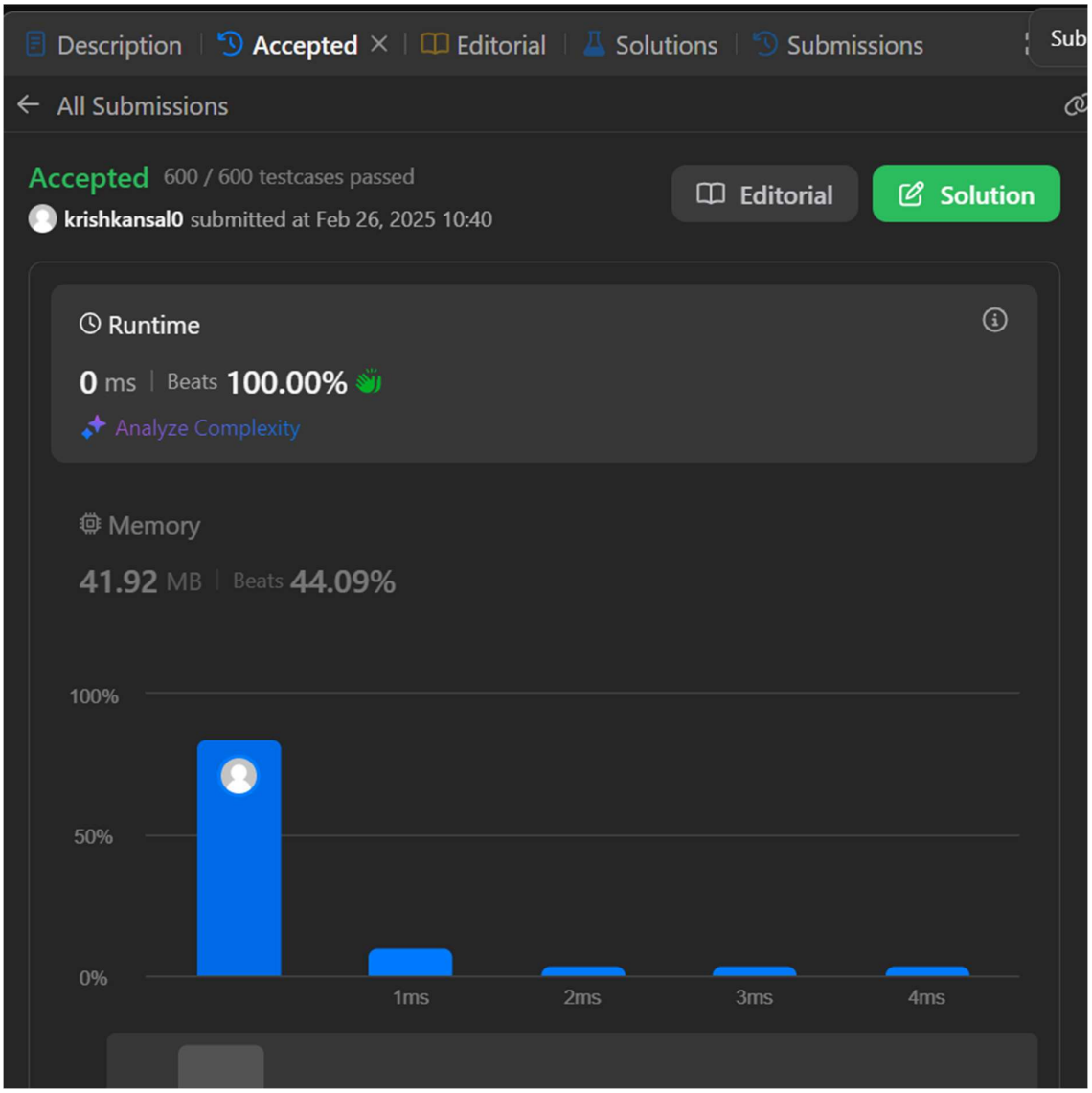
Longest Nice Substring: -

```java
class Solution {
    public String longestNiceSubstring(String s) {
        if (s.length() < 2) return "";

        for (int i = 0; i < s.length(); i++) {
            char c = s.charAt(i);
            if (s.contains(Character.toString(Character.toUpperCase(c))) &&
                s.contains(Character.toString(Character.toLowerCase(c)))) {
                continue;
            }

            String left = longestNiceSubstring(s.substring(0, i));
            String right = longestNiceSubstring(s.substring(i + 1));


            return left.length() >= right.length() ? left : right;
        }

        return s;
    }

    public static void main(String[] args) {
        Solution sol = new Solution();
        System.out.println(sol.longestNiceSubstring("YazaAay"));
    }
}
```



Reverse Bits: -



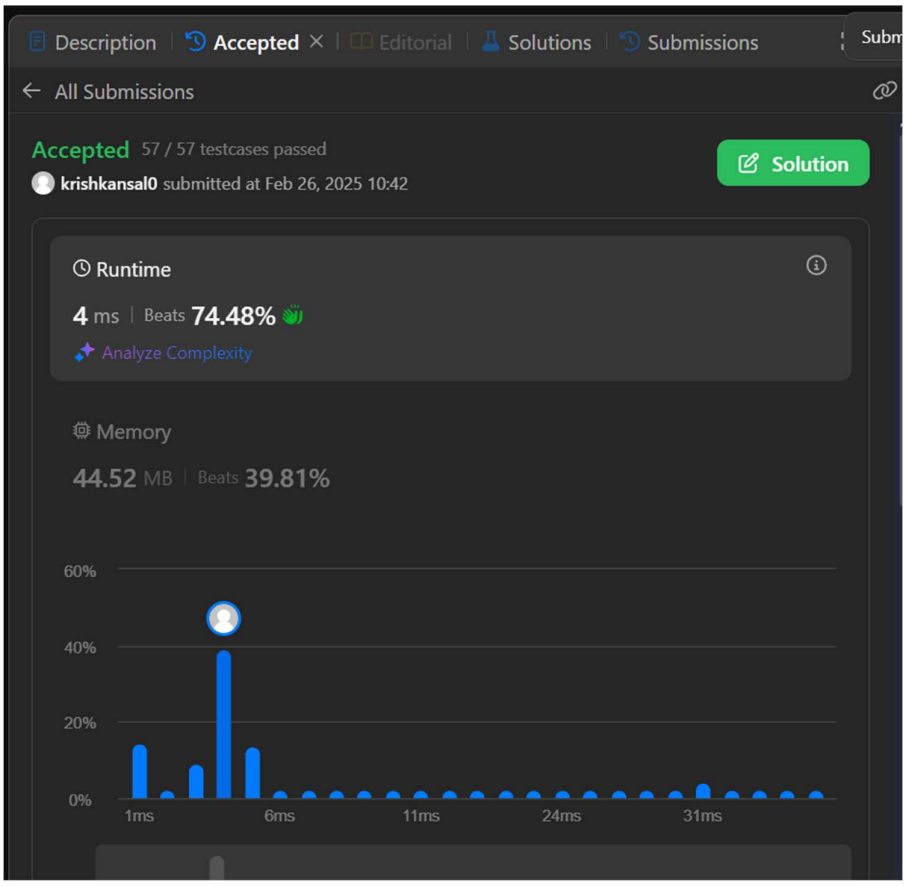


Super pow

Code

Java Auto

```java
class Solution {
    static final int MOD = 1337;

    private int modPow(int x, int y) {
        x %= MOD;
        int result = 1;
        while (y > 0) {
            if (y % 2 == 1) {
                result = (result * x) % MOD;
            }
            x = (x * x) % MOD;
            y /= 2;
        }
        return result;
    }

    public int superPow(int a, int[] b) {
        int result = 1;
        for (int digit : b) {
            result = (modPow(result, 10) * modPow(a, digit)) % MOD;
        }
        return result;
    }
}
```

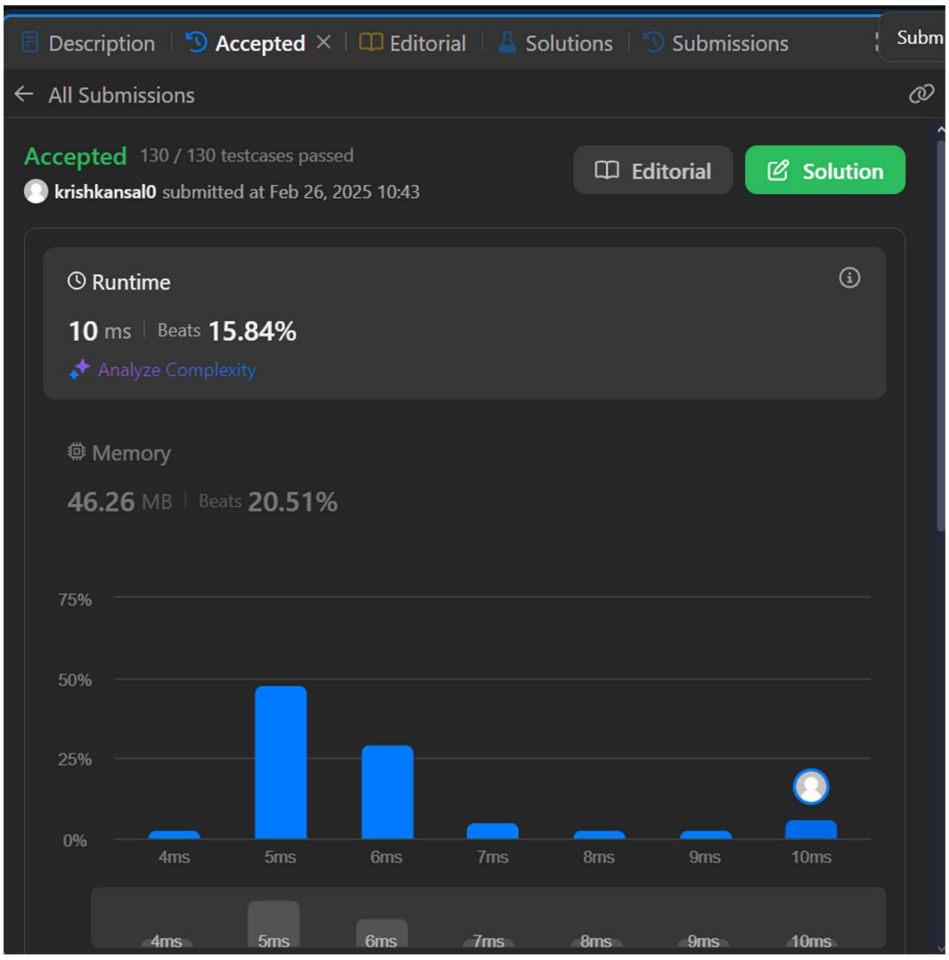


Search a 2D Matrix II

Code

Java Auto

```java
class Solution {
    public boolean searchMatrix(int[][] matrix, int target) {
        for(int i=0;i< matrix.length;i++){
            for(int j=0;j< matrix[0].length;j++){
                if(matrix[i][j]>target){
                    break;
                }
                if(matrix[i][j]==target){
                    return true;
                }
            }
        }
        return false;
    }
}
```



Maximum Subarray

Code

Java | Auto

```java
class Solution {
    public int maxSubArray(int[] nums) {
        int maxSum = nums[0], currentSum = nums[0];
        for (int i = 1; i < nums.length; i++) {
            currentSum = Math.max(nums[i], currentSum + nums[i]);
            maxSum = Math.max(maxSum, currentSum);
        }
        return maxSum;
    }
}
```