

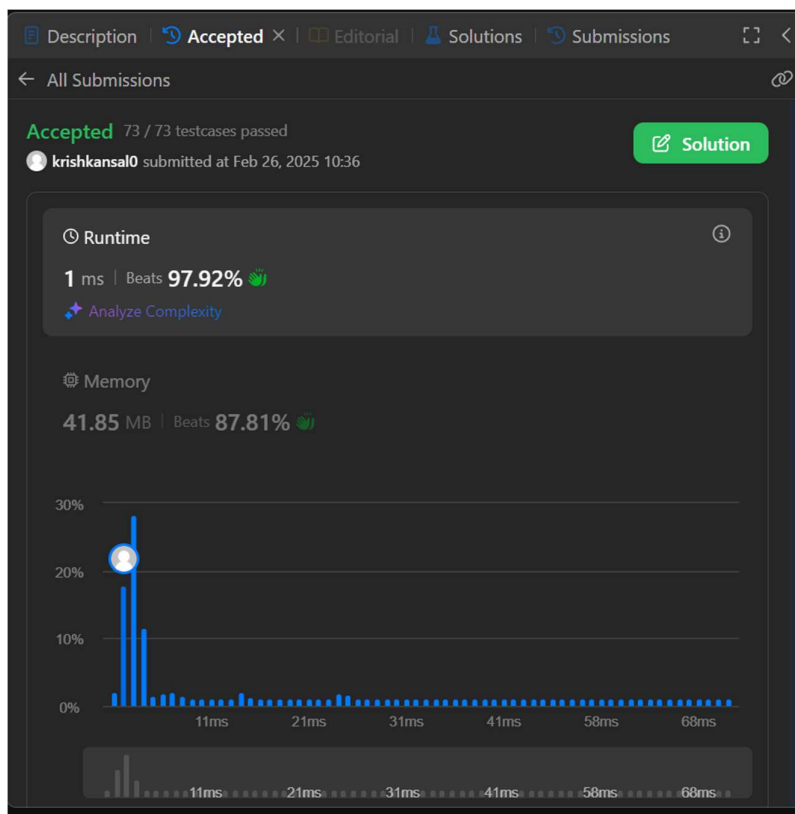
Krish Aggarwal

22BCS16370

Longest Nice Substring: -

```
Code
Java Auto
1 class Solution {
2     public String longestNiceSubstring(String s) {
3         if (s.length() < 2) return "";
4
5         for (int i = 0; i < s.length(); i++) {
6             char c = s.charAt(i);
7             if (s.contains(Character.toString(Character.toUpperCase(c))) &&
8                 s.contains(Character.toString(Character.toLowerCase(c)))) {
9                 continue;
10            }
11
12            String left = longestNiceSubstring(s.substring(0, i));
13            String right = longestNiceSubstring(s.substring(i + 1));
14
15
16            return left.length() >= right.length() ? left : right;
17        }
18
19        return s;
20    }
21
22    public static void main(String[] args) {
23        Solution sol = new Solution();
24        System.out.println(sol.longestNiceSubstring("YazaAay"));
25    }
26 }
```

Saved Ln 1, Col 1



Reverse Bits: -

```
Ctrl Enter

Java Auto

1 public class Solution {
2     public int reverseBits(int n) {
3         int result = 0;
4         for (int i = 0; i < 32; i++) {
5             result = (result << 1) | (n & 1);
6             n >>= 1;
7         }
8         return result;
9     }
10 }
11
```

Description | Accepted × | Editorial | Solutions | Submissions

← All Submissions

**Accepted** 600 / 600 testcases passed

krishkansal0 submitted at Feb 26, 2025 10:40

Editorial Solution

**Runtime**

0 ms | Beats 100.00% 🍃

Analyze Complexity

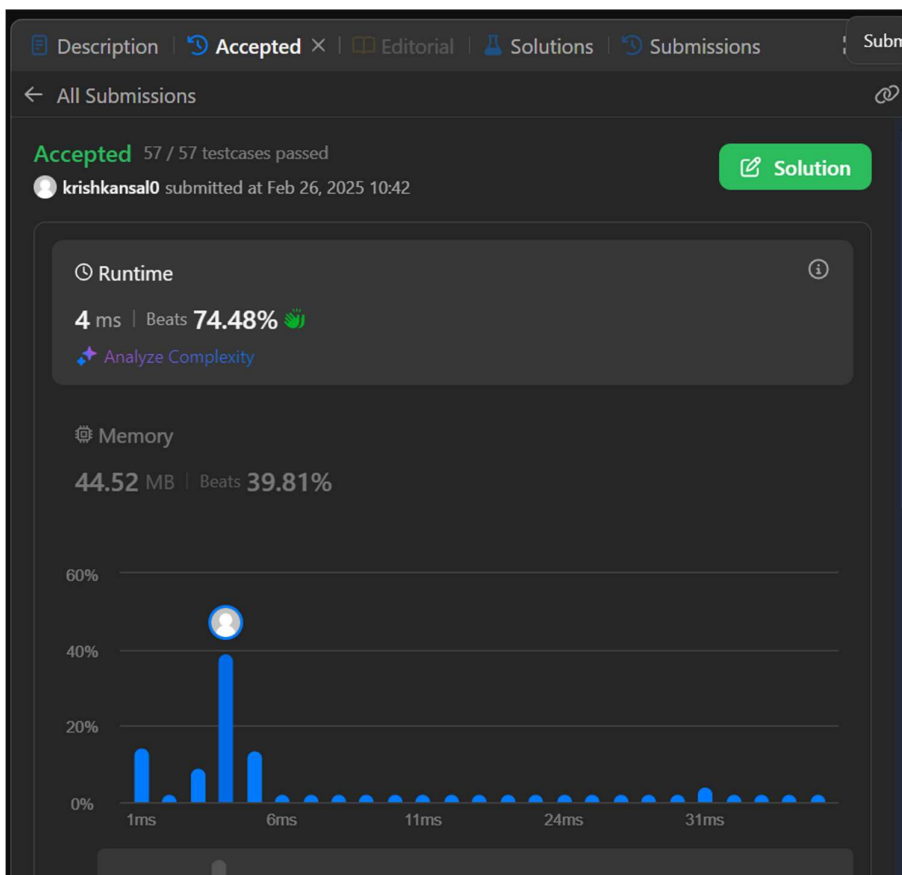
**Memory**

41.92 MB | Beats 44.09%

Performance Metric	User	1ms	2ms	3ms	4ms
Runtime (ms)	0	1	2	3	4

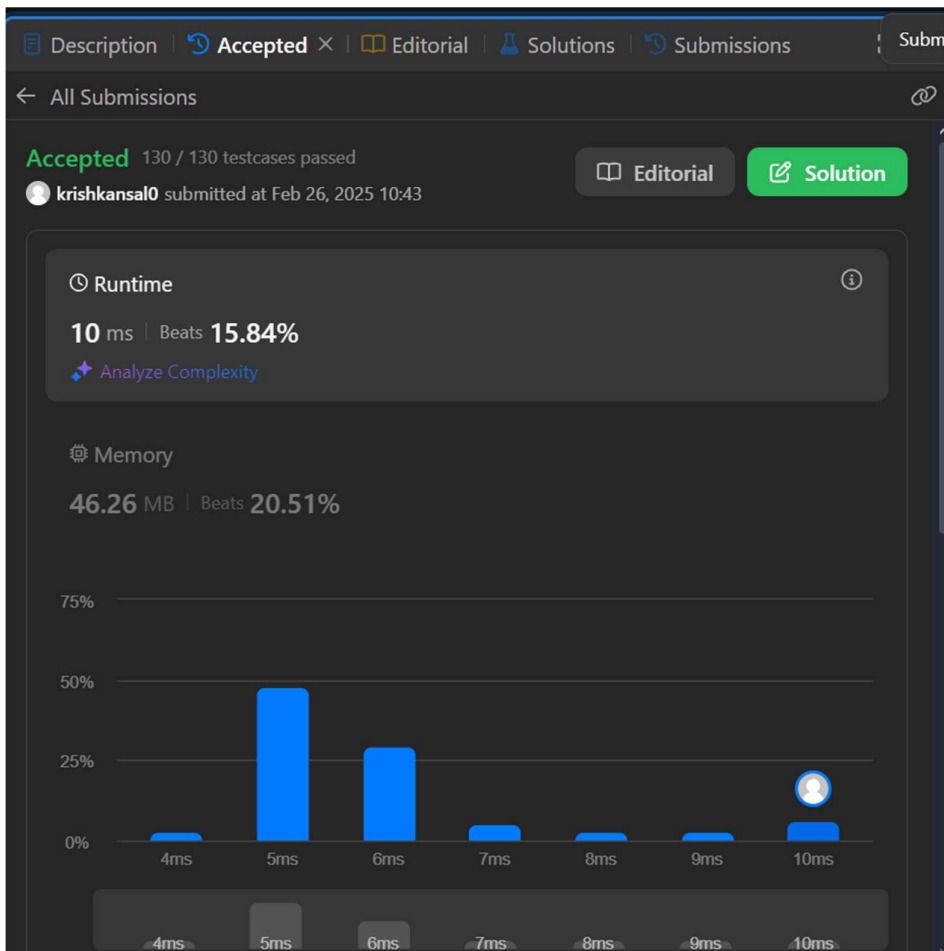
## Super pow

```
Code
Java Auto
1 class Solution {
2     static final int MOD = 1337;
3
4     private int modPow(int x, int y) {
5         x %= MOD;
6         int result = 1;
7         while (y > 0) {
8             if (y % 2 == 1) {
9                 result = (result * x) % MOD;
10            }
11            x = (x * x) % MOD;
12            y /= 2;
13        }
14        return result;
15    }
16
17    public int superPow(int a, int[] b) {
18        int result = 1;
19        for (int digit : b) {
20            result = (modPow(result, 10) * modPow(a, digit)) % MOD;
21        }
22        return result;
23    }
24 }
```



## Search a 2D Matrix II

```
</> Code
Java Auto
1 class Solution {
2     public boolean searchMatrix(int[][] matrix, int target) {
3         for(int i=0;i< matrix.length;i++){
4             for(int j=0;j< matrix[0].length;j++){
5                 if(matrix[i][j]>target){
6                     break;
7                 }
8                 if(matrix[i][j]==target){
9                     return true;
10                }
11            }
12        }
13        return false;
14    }
15 }
```



## Maximum Subarray

```
</> Code
Java Auto

1 class Solution {
2     public int maxSubArray(int[] nums) {
3         int maxSum = nums[0], currentSum = nums[0];
4         for (int i = 1; i < nums.length; i++) {
5             currentSum = Math.max(nums[i], currentSum + nums[i]);
6             maxSum = Math.max(maxSum, currentSum);
7         }
8         return maxSum;
9     }
10 }
11
```

Description | Accepted X | Editorial | Solutions | Submissions

← All Submissions

**Accepted** 210 / 210 testcases passed

krishkansal0 submitted at Feb 26, 2025 10:45

Editorial Solution

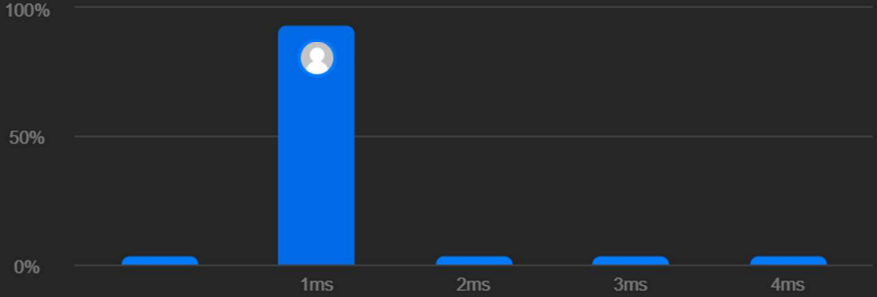
⌚ Runtime ⓘ

1 ms | Beats 99.53% 🌿

🔍 Analyze Complexity

💻 Memory

57.44 MB | Beats 13.31%



Time Interval	Percentage
1ms	~99.53%
2ms	~0%
3ms	~0%
4ms	~0%