Question: Longest Nice substring

class Solution {

public String longestNiceSubstring(String s) {

int n = s.length();

int max = 0;

String res = "";

for(int i = 0; i < n; i++){

for(int j = i; j < n; j++){

StringBuilder sb = new StringBuilder();

for(int k = i; k <= j; k++){

sb.append(s.charAt(k));

}

if(checkIfNice(sb)){

if(j - i + 1 > max){

max = j - i + 1;

res = sb.toString();

}

}

}

}

return res;

}

public boolean checkIfNice(StringBuilder s){

HashSet<Character> set = new HashSet<>();

for(int i = 0; i < s.length(); i++){

set.add(s.charAt(i));

}

for(int i = 0; i < s.length(); i++){

if(s.charAt(i) >= 65 && s.charAt(i) <= 90){

if(!set.contains((char)(s.charAt(i) + 32))){

return false;

}

}

if(s.charAt(i) >= 97 && s.charAt(i) <= 122){

if(!set.contains((char)(s.charAt(i) - 32))){

return false;

}

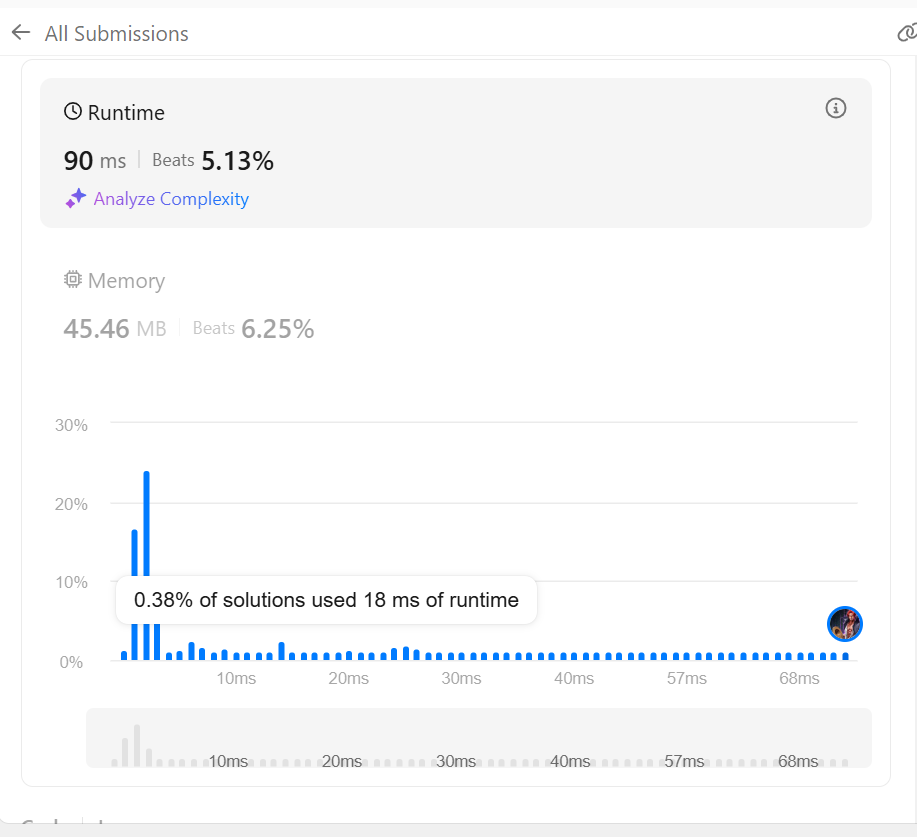
}

}

return true;

}

}



Question: Maximum subarray

class Solution {

public int maxSubArray(int[] nums) {

int res = 0;

int currSum = 0;

for (int i = 0; i < nums.length; i++) {

currSum += nums[i];

if (currSum < 0) {

currSum = 0;

} else {

res = Math.max(currSum, res);

}

}

if (res == 0) {

res = Integer.MIN\_VALUE;

for (int i = 0; i < nums.length; i++) {

res = Math.max(res, nums[i]);

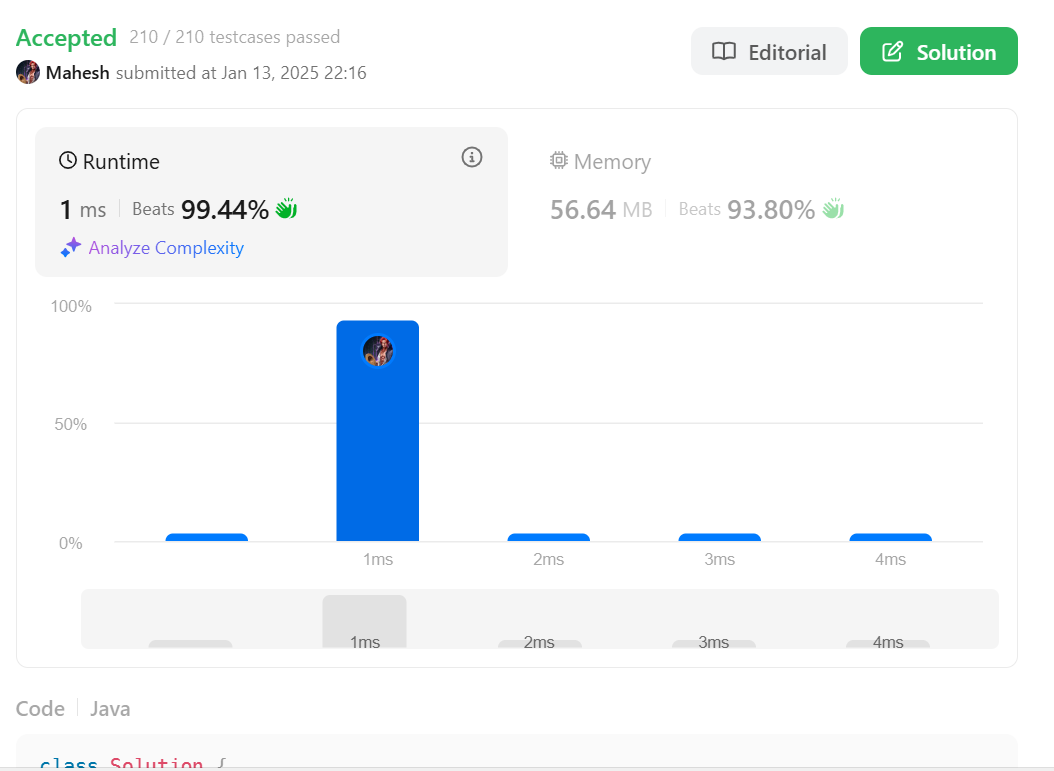
}

}

return res;

}

}



Question : Search a 2D matrix II  
public class Solution {

public boolean searchMatrix(int[][] matrix, int target) {

if(matrix == null || matrix.length < 1 || matrix[0].length <1) {

return false;

}

// Starting from top right corner

int col = matrix[0].length-1;

int row = 0;

while(col >= 0 && row <= matrix.length-1) {

if(target == matrix[row][col]) {

return true;

} else if(target < matrix[row][col]) {

col--;

} else if(target > matrix[row][col]) {

row++;

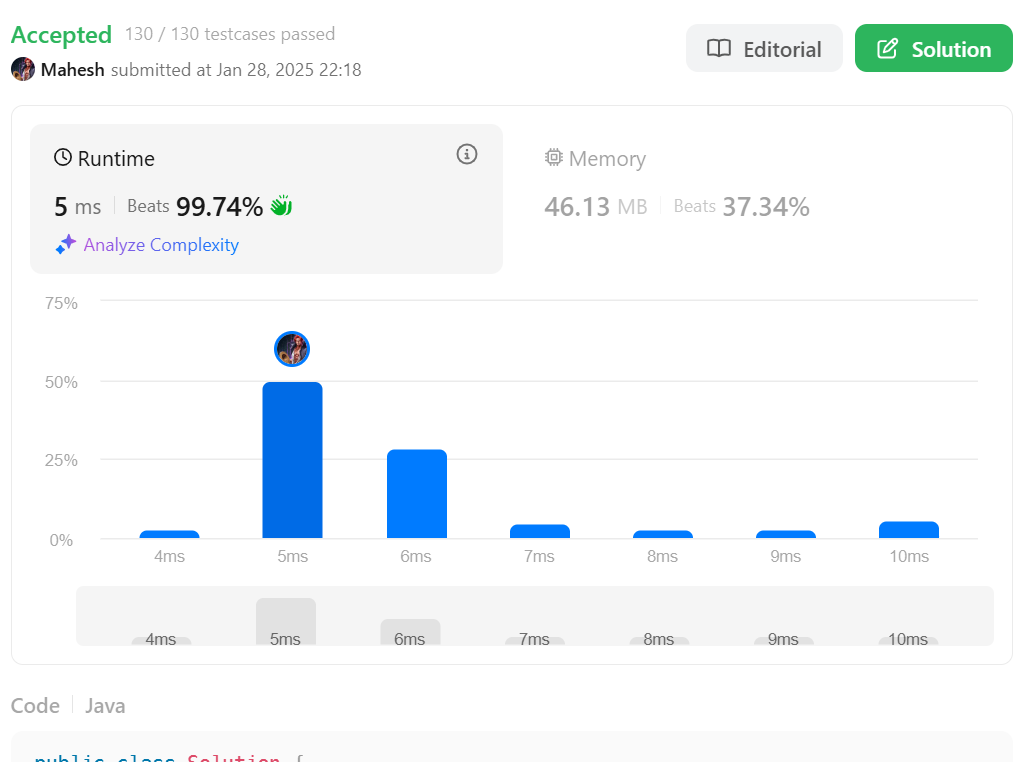
}

}

return false;

}

}



Question: Merge Sorted array

class Solution {

public void merge(int[] nums1, int m, int[] nums2, int n) {

int end = nums1.length - 1;

int i = m - 1;

int j = n - 1;

while (end >= 0 && i >= 0 && j >= 0) {

if (nums1[i] <= nums2[j]) {

nums1[end] = nums2[j];

j--;

end--;

} else if (nums1[i] > nums2[j]) {

nums1[end] = nums1[i];

i--;

end--;

}

}

while (j >= 0) {

nums1[end] = nums2[j];

j--;

end--;

}

}

}

