[https://leetcode.com/problems/merge-strings-alternately/](https://leetcode.com/problems/merge-strings-alternately/" \t "_blank) заводим один указатель, (пока меньше 1 строки или пока меньше второй) (в цикле проверяем в какой именно строке находимся и добавляем в ответ) Используем StringBuilder

class Solution {

    public String mergeAlternately(String word1, String word2) {

       StringBuilder  res = new StringBuilder("");

       int len = word1.length() + word2.length();

       for (int i=0; i < len; i++){

           if (i < word1.length()) res.append(word1.charAt(i));

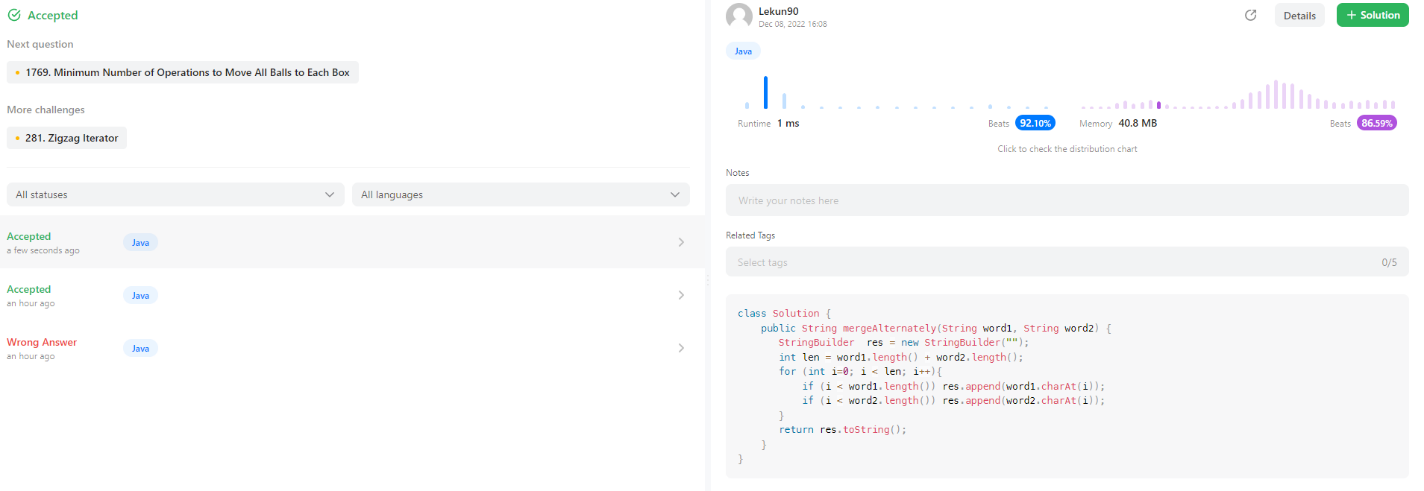
           if (i < word2.length()) res.append(word2.charAt(i));

       }

       return res.toString();

    }

}



[https://leetcode.com/problems/remove-element/](https://leetcode.com/problems/remove-element/" \t "_blank) решается как moveZeroes

class Solution {

    public int removeElement(int[] nums, int val) {

        if (nums.length == 0) return 0;

        int size = 0;

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

            if (nums[i] != val){

                nums[size] = nums[i];

                size++;

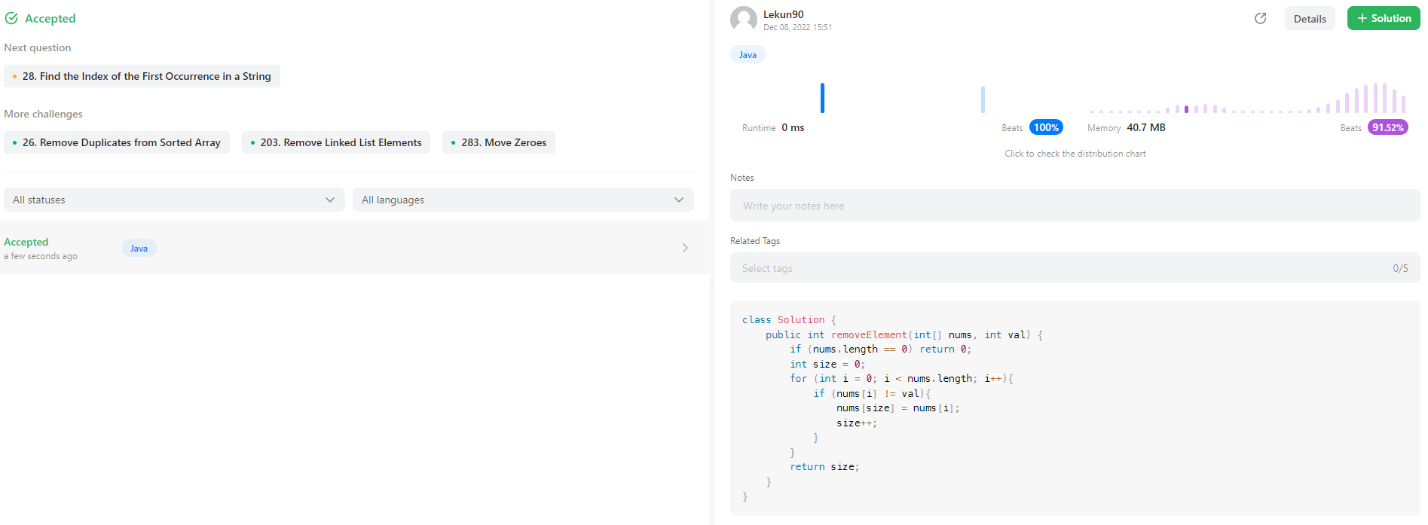
            }

        }

        return size;

    }

}



<https://leetcode.com/problems/merge-sorted-array/> новый массив создавать и сортировать не надо

class Solution {

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

        int poz1 = m-1, poz2 = n-1, index = m + n - 1;

        while (poz2 >= 0){

            if (poz1 >= 0 && nums1[poz1] > nums2[poz2]){

                nums1[index--] = nums1[poz1--];

            }

            else {

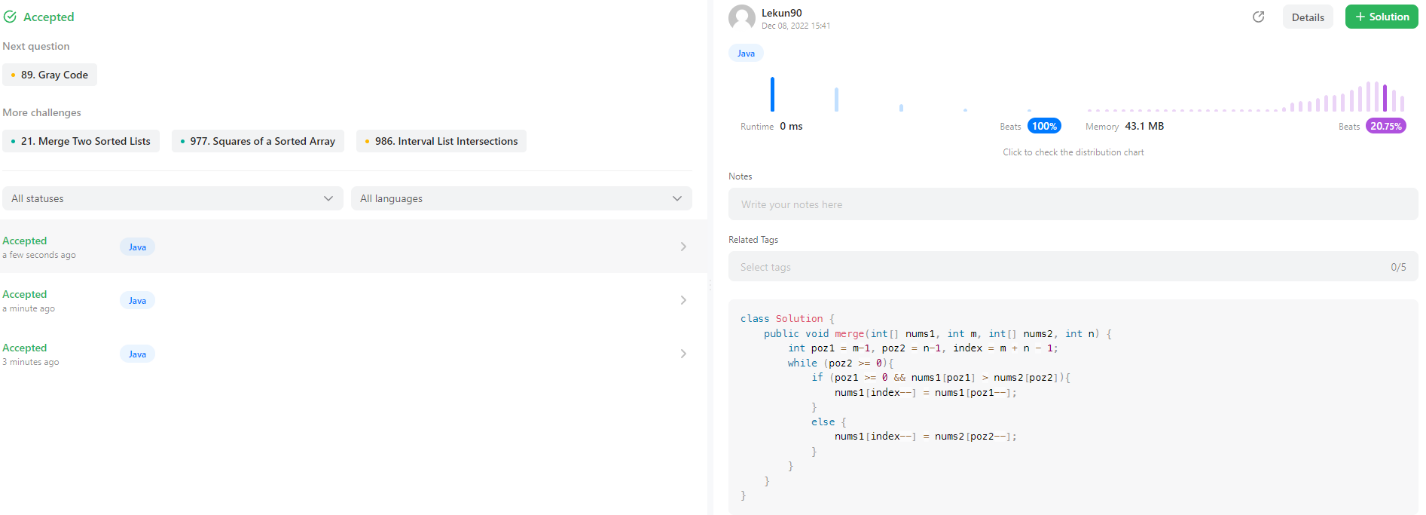
                nums1[index--] = nums2[poz2--];

            }

        }

    }

}



<https://leetcode.com/problems/reverse-words-in-a-string/> (в записи есть решение) заводим переменные end (указатель на конец слова, двигаем пока не достигли символа) заводим start = end - 1 и двигаем пока не дойдем до пробела.

class Solution {

    public String reverseWords(String s) {

        String[] words = s.split(" ");

        StringBuilder result = new StringBuilder(words.length);

        for (int i = words.length - 1; i >= 0; i--){

            if (words[i].length() != 0){

                if (result.length() != 0) result.append(" ");

                result.append(words[i]);

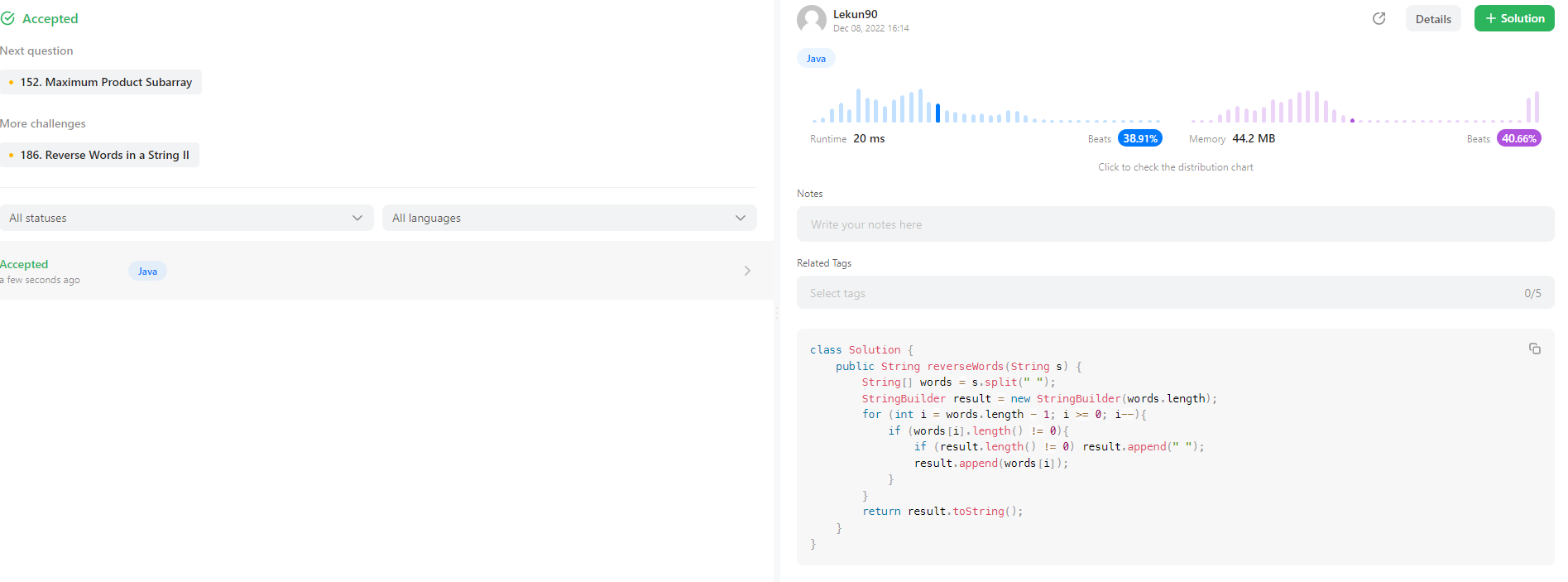
            }

        }

        return result.toString();

    }

}



<https://leetcode.com/problems/valid-palindrome/> нужно пропустить все не буквы или цифры. Есть метод Character.isLetterOrDigit(char ch). Принимает символ и говорит это буква или цифра. Заводим указатель left = 0; right = string.length() - 1. Перед тем как проверить двигаем их пока не встанем в символ. Метод replace под запретом, строку создавать новую или менять исходную нельзя

class Solution {

    public boolean isPalindrome(String s) {

        for (int i = 0, j = s.length() - 1; i < j; i++, j--){

            while(i < j && !Character.isLetterOrDigit(s.charAt(i))){

                i++;

            }

            while(i < j && !Character.isLetterOrDigit(s.charAt(j))){

                j--;

            }

            if(Character.toLowerCase(s.charAt(i)) != Character.toLowerCase(s.charAt(j))) return false;

        }

        return true;

    }

}

