ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Жадные алгоритмы»

Выполнил работу

Старцев Иван

Академическая группа №j3110

Принято

Должность, Владислав Вершинин

Санкт-Петербург

2024

**Структура отчёта:**

1. Введение

Цель работы – Реализовать жадный алгоритм.

Задачи:

* Выбрать задачу на leetcode
* Понять где здесь необходим жадный алгоритм
* Написать годный код
* Получить галочку на leetcode

1. Теоретическая подготовка

Для реализации данной задачи требовалось небольшое количество знаний математики, а для самого алгоритма нужно было знать только цикл

1. Реализация

Первым этапом был выбор задачи на leetcode.

Вторым этапом я обдумывал решение самым жадным способом.

Третьим этапом была проверка моей теории на мини тестах

Теория заключалась в том, что как минимум придется сделать столько шагов какое значение имеет первый элемент, а каждый стоит обрабатывать основываясь на значение на предыдущего.

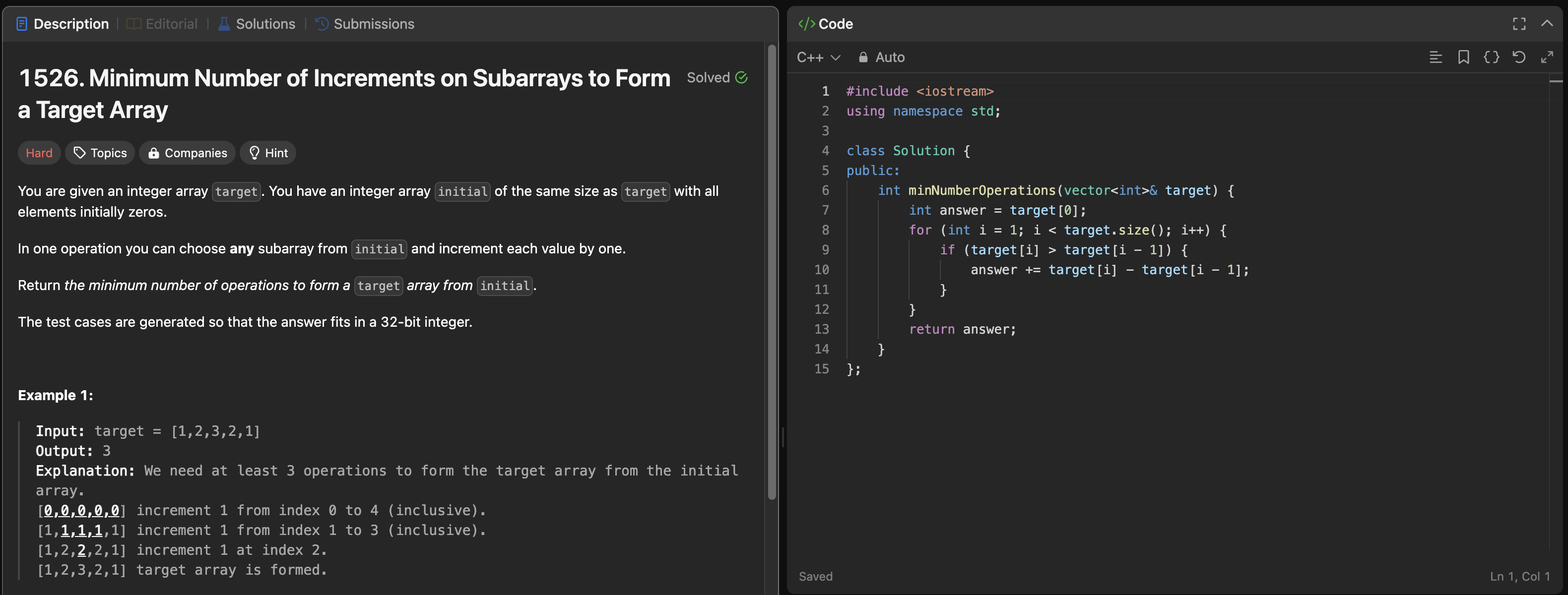
Последним этапом шла реализация кода и проверка его leetcode.

1. Экспериментальная часть

Сложность по памяти алгоритма – O(1)

Временная сложность алгоритма – O(N)

Док-во зачета на Leetcode и сам код показаны на изображении №1 ниже.



Изображение №1 – Результат на LeetCode

1. Заключение

В ходе выполнения работы мною был реализован жадный алгоритм для подсчета кол-ва трансформаций для получения из массива нулей целевой массив. Цель работы была достигнута путём обдумывания жадного алгоритма и его тестов на маленьких примерах. Полученные результаты также совпадают с теоретическими оценками сложности алгоритма.

В качестве дальнейших рассуждений предложить нечего, т.к в любом случае придется посмотреть каждое значение целевого массива, а наше решение реализовано за O(N)

1. Приложения

ПРИЛОЖЕНИЕ А

Листинг кода файла main.cpp

#include <iostream>

using namespace std;

class Solution {

public:

int minNumberOperations(vector<int>& target) {

int answer = target[0];

for (int i = 1; i < target.size(); i++) {

if (target[i] > target[i - 1]) {

answer += target[i] - target[i - 1];

}

}

return answer;

}

};