ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Minimum Number of Increments on Subarrays to Form a Target Array(leetcode)»

Выполнил работу

Крапивина Яна

Академическая группа №J3112

Принято

Должность, звание Фамилия Имя преподавателя

Дунаев Максим, практик

Санкт-Петербург

2024

**Структура отчёта:**

1. Введение

Цель работы: разработка алгоритма, который вычисляет минимальное количество операций, необходимых для преобразования массива initial, состоящего из нулей, в заданный массив target. Операции заключаются в выборе подмассива и увеличении каждого его элемента на единицу.

Задачи:

* Анализ входных данных
* Определение операции(инкрементирование всех элементов выбранного подмассива на единицу)
* Разработка жадного алгоритма

1. Теоретическая подготовка

Используемые типы данных:

* vector<int>: для представления массива target
* int: для хранения переменных, таких как operations, previous и current

1. Реализация

1)Анализ задачи:

* Задача требует найти минимальное количество инкрементов на подмассивах для формирования целевого массива
* Вернуть нужно минимальное количество операций, необходимых для формирования массива target из initial

В данной задаче о преобразовании массива из нулей в заданный массив целых чисел с использованием инкрементов на подмассивах, ***жадный алгоритм*** оказывается наиболее подходящим потому, что нет необходимости возвращаться к предыдущим решениям, выбираем локально оптимальное решение, можно решить задачу за О(n). Другие подходы, такие как динамическое программирование, могут быть избыточными и сложными, учитывая, что задача не требует учета сложных зависимостей между элементами.

2)Реализация алгоритма:

* Объявление переменных:

int operations = 0;

int previous = 0;

* Итерация по массиву target:

for (int current : target) {

if (current > previous) {

operations += (current - previous);

* Расчет операций:

operations += (current - previous);

* Обновление предыдущего значения:

previous = current;

* Возврат результата:

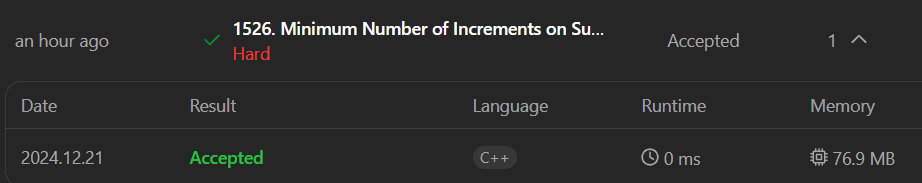
return operations;

1. Экспериментальная часть

Подсчёт по памяти: O(1), поскольку количество используемых переменных не зависит от размера входного вектора. Не создаём дополнительных структур данных, которые бы зависели от размера входных данных.

Подсчёт асимптотики: O(n) , где n — это количество элементов в векторе target.

Все тесты пройдены:



1. Заключение

В ходе выполнения работы был разработан алгоритм для вычисления минимального количества операций, необходимых для преобразования массива, состоящего из нулей, в целевой массив target с использованием инкрементов на подмассивах. Алгоритм реализован на языке C++ и использует итеративный подход для подсчета операций, основанный на разнице между текущим и предыдущим элементами массива.

Цель работы была достигнута путем анализа структуры массива target, что позволило эффективно определить количество необходимых инкрементов. Полученные результаты были протестированы на различных тестовых случаях, что подтвердило правильность и эффективность алгоритма. Временная сложность алгоритма составляет O(n) , где n — это длина массива target, что соответствует ожиданиям для данной задачи.

1. Приложения

ПРИЛОЖЕНИЕ А

Листинг кода

#include <vector>

using namespace std;

class Solution {

public:

int minNumberOperations(vector<int>& target) {

int operations = 0;

int previous = 0;

for (int current : target) {

if (current > previous) {

operations += (current - previous);

}

previous = current;

}

return operations;

}

};