ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 7

«Решение задачи [Minimum Number of Increments on Subarrays to Form a Target Array](https://leetcode.com/problems/minimum-number-of-increments-on-subarrays-to-form-a-target-array/)»

Выполнил работу

Московкин Александр Николаевич

Академическая группа №J3111

Ментор, Владислав Константинович

Санкт-Петербург

2024

1. **Введение**

**Цель работы:** Разработать алгоритм, который минимизирует количество операций, необходимых для преобразования массива *initial*, состоящего из нулей, в массив *target* с заданными значениями.

**Задачи лабораторной работы:**

1. Понять и описать подход к минимизации операций.
2. Реализовать алгоритм с минимальной временной и пространственной сложностью.
3. Провести тестирование и оценить эффективность.
4. **Теоретическая подготовка**

**Суть задачи:**

Дан массив *target*, инициализированный массив *initial* состоит из нулей. За одну операцию можно увеличить на 1 все элементы некоторого подмассива. Требуется минимизировать количество операций, необходимых для того, чтобы массив *initial* стал равен *target*.

**Основной подход:**

Каждое увеличение значения в массиве *target* требует дополнительной операции.

Если значение уменьшается или остаётся равным предыдущему, дополнительных операций не требуется.

**Ключевая идея:**

Для минимизации операций необходимо учитывать только разницу между текущим элементом и предыдущим. Если текущее значение больше предыдущего, добавляется разница, иначе ничего не происходит.

1. **Реализация**

Алгоритм разработан с использованием итерационного подхода:

1. Инициализируются переменные:

*operations* — счётчик операций (изначально равен 0).

*prevWeight* — вес предыдущего элемента (изначально 0).

2. Для каждого элемента массива *target* вычисляется разница с предыдущим значением:

Если текущее значение больше *prevWeight*, разница добавляется к *operations*.

Обновляется значение *prevWeight*.

3. Итоговое значение *operations* возвращается как результат.

1. **Экспериментальная** **часть**

На данном этапе произведён анализ используемой памяти и вычислительной сложности для реализации алгоритма минимизации разрезов строки на палиндромы.

**Подсчёт памяти**

* Используется *O(1)* дополнительной памяти для переменных.

**Подсчёт асимптотики**

* Один проход по массиву *target*, *O(n)*, где *n* — длина массива.

1. **Заключение**

В ходе выполнения работы был разработан алгоритм, минимизирующий количество операций для формирования массива *target*.

**Преимущества:**

* Простая реализация с линейной временной сложностью.
* Низкие затраты памяти, так как алгоритм работает на месте.

**Результаты:**

* Алгоритм успешно протестирован на различных примерах, включая массивы с возрастанием, убыванием и произвольным чередованием значений.

1. **Приложения**

Полный исходный код программы

ПРИЛОЖЕНИЕ А

Листинг кода файла lab7.cpp

