ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 6

«Динамическое программирование. Решение задачи arithmetic slices II subsequence»

Выполнил работу

Кащеев Максим Николаевич

Академическая группа J3113

Принято

Иван Владимирович Ходненко

Санкт-Петербург

2024

**Структура отчёта:**

1. Введение

Целью выполнения данной работы является понимание, почему некоторые задачи оптимально решать динамическим программированием, на примере конкретной задачи.

1. Теоретическая подготовка

Для выполнения данной лабораторной работы требуется понимание подхода динамического программирования для решения задач. Также необходимо минимальное понимание классов, типов данных int, vector, unordered\_map, long long.

1. Реализация

В ходе выполнения данной лабораторной работы я реализовал алгоритм с асимптотической сложностью O(N^2) в среднем случае, однако при возникновении коллизий в хеш таблице (map) сложность может теоретически опускаться до N^3, но поскольку такие случаи на практике минимизированы алгоритмом хеширования, то такой случай встречается чрезвычайно редко, для практически всех реальный случаев сложность будет N^2. Алгоритм раскладывает последовательность на подпоследовательности по заданной метрике. Использование памяти составляет O(N^2) – за N^2 итераций цикла выполняются простые операции внутри циклов, которые незначительно влияют на память, наибольшее влияние оказывает работа с map, от O(1) в общем случае и до O(k) в худшем, где k – количество элементов в map.

1. Экспериментальная часть

Изображение выглядит как текст, программное обеспечение, Значок на компьютере, число

Автоматически созданное описание

Изображение №1 – Реализация алгоритма на сайте leetcode

Изображение выглядит как текст, снимок экрана, Шрифт, число

Автоматически созданное описание

Изображение №2 – указание сложности алгоритма на сайте leetcode

1. Заключение

В ходе выполнения работы мною был реализован алгоритм на основе динамического программирования для поиска всех возможных подпоследовательностей длинной >= 3 для исходного массива. Цель работы была достигнута, так как, исходя из данного вида задачи выгоднее всего разбить её на подзадачи, которые, в отличие, например, от рекурсии, не будут делать большое количество итераций, зависящих напрямую от длинны исходного массива. Также задача и решение относятся к динамическому программированию, поскольку на итерациях используются значения вычисленные ранее, и не приходится несколько раз считать одни и те же значения, так как они хранятся в отдельном массиве.

1. Приложения

В этом разделе вам необходимо указать всю дополнительную информацию, которая не вошла в основной текст, например полный исходный код программы. Пример:

ПРИЛОЖЕНИЕ А

Листинг кода файла main.cpp

class Solution {

public:

    int numberOfArithmeticSlices(vector<int>& nums) {

        int totalCount = 0;

        vector<unordered\_map<long long, int>> dp(nums.size());

        for (int i = 0; i < size(nums); ++i)

        {

            for (int j = 0; j < i; ++j)

            {

                long long diff = static\_cast<long long>(nums[i]) - nums[j];

                int countAtJ = dp[j][diff];

                dp[i][diff] += countAtJ + 1;

                totalCount += countAtJ;

            }

        }

        return totalCount;

    }

};