ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО»

Отчёт по лабораторной работе № 6

«85. Maximal Rectange»

Выполнил работу

Максудов Кирилл

Академическая группа J3111

Принято

Практик Вершинин Владислав

Санкт-Петербург

2024

Введение

Цель работы: дана матрица, определить размер наибольшего прямоугольника содержащего только 1 и вернуть его площадь.

Задачи:

1. Написать алгоритм
2. Подсчитать память
3. Измерить время работы алгоритма
4. Объяснить зачем нужно ДП
5. Теоретическая подготовка

Использовал векторы, stack, int, string и тд.

1. Реализация

Создаем вектор height длинны количества столбцов, в нем обновляем с каждой строкой непрерывные высоты для данного столбца и передаем в функцию largestRectangleArea, где ищем максимальные длины по строкам для каждого столбца как самого высокого. Получаем размер прямоугольника, сравниваем с максимальным и продолжаем.

1. Экспериментальная часть

Подсчет по памяти(только для циклов и сложных структур):

Пространство, необходимое для std::vector<int> heights, std::vector<int> left, std::vector<int> right и std::stack<int> index\_stack равно n => пространственная сложность составляет O(n).

Подсчёт асимптотики (только для циклов и сложных структур):

В maximalRectangle проходим по каждому элементу матрицы ровно 1 раз => сложность составляет O(n\*m) где n и m – высота и ширина матрицы

В largestRectangleArea проходим последовательно по строке несколько раз и так для каждой строки => сложность тоже O(n\*m)

Общая сложность O(n\*m)

Изображение №1 – График зависимости времени от размера матрицы

Теоретически заданная сложность задачи составляет O(n\*m) и менее т.к мы не всегда будем проходить весь height. Для тестирования алгоритма была собрана статистика, приведенная в таблице №1.

Таблица №1 - Подсчёт сложности реализованного алгоритма

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Размер матрицы | 10x10 | 100x100 | 250x250 | 500x500 |
| Расчетное время, с | 0.00003700 | 0,0037 | 0,023125 | 0,0925 |
| Время, c | 0.00003700 | 0.00145980 | 0.00461320 | 0.01473710 |

1. Заключение

В ходе выполнения работы мною был реализован алгоритм поиска прямоугольника максимального размера в матрице. В этой задаче используется ДМ для сохранения и изменения значений в векторе height и использования их в дальнейшем, таким образом мы не пересчитываем их каждый раз заново

ПРИЛОЖЕНИЕ А

Листинг кода файла lab-6.cpp

Изображение выглядит как текст, снимок экрана

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, программное обеспечение, Операционная система

Автоматически созданное описание

Изображение выглядит как текст, снимок экрана, дисплей, программное обеспечение

Автоматически созданное описание

ПРИЛОЖЕНИЕ B

Результат на сайте leetcode.com

Изображение выглядит как снимок экрана, текст, программное обеспечение, Мультимедийное программное обеспечение

Автоматически созданное описание