Лабораторная работа № 7 по курсу дискретного анализа: Динамическое программирование

Выполнил студент группы 08-308 МАИ *Попов Николай*.

Условие

Задан прямоугольник с высотой n и шириной m, состоящий из нулей и единиц. Найдите в нём прямоугольник наибольшой площади, состоящий из одних нулей.

Метод решения

Вычисление "высоты": Сначала создаётся вспомогательная матрица height, в которой для каждой ячейки (i, j) хранится количество последовательных единиц в столбце сверху вниз до этой ячейки, если в исходной матрице на этом месте стоит 0.

Поиск максимальной площади для каждой строки: Для каждой строки вычисляется максимально возможная площадь прямоугольника, причём в расчёт берутся только прямоугольники, которые "заканчиваются"в этой строке. Используется стек, чтобы сохранять индексы столбцов, которые потенциально могут участвовать в формировании прямоугольника. Если следующая "высота"больше или равна последней в стеке, индекс столбца добавляется в стек. Если меньше, то из стека удаляются индексы, пока не будет найдена меньшая высота, и при этом вычисляется площадь для возможного прямоугольника.

Вычисление площади по оставшимся индексам в стеке: После прохода по всем столбцам строки, для индексов, оставшихся в стеке, также вычисляется площадь, так как они могут образовывать прямоугольники, простирающиеся до конца строки.

Описание программы

1. Инициализируется матрица `height`, где для каждой ячейки хранится количество последовательных единиц в соответствующем столбце:

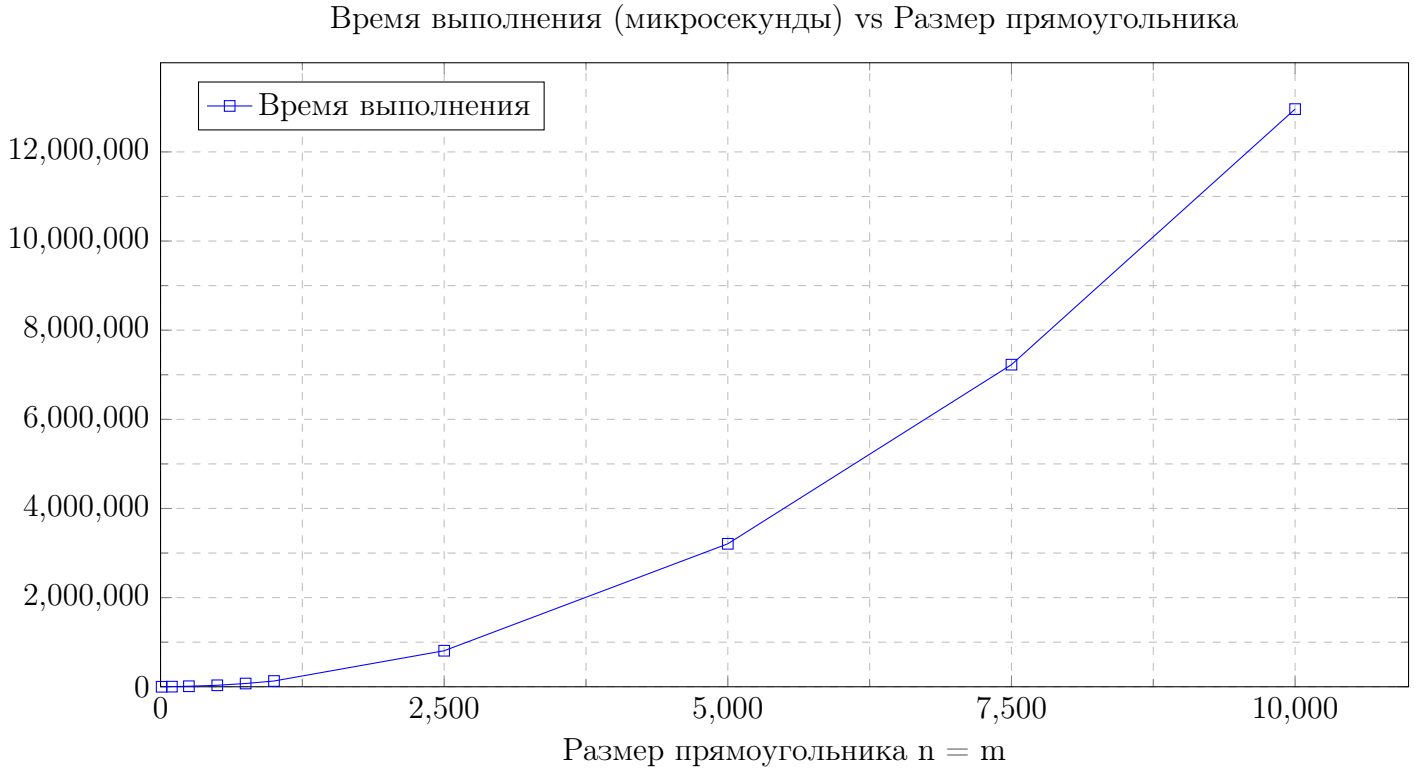
```
vector<vector<int> > height(n, vector<int>(m, 0));
for (int i = 0; i < n; ++i) {
    for (int j = 0; j < m; ++j) {
        if (matrix[i][j] == 0) {
            height[i][j] = (i == 0) ? 0 : height[i - 1][j] + 1;
        }
    }
}
```

2. Для каждой строки матрицы высоты ищется максимальная площадь прямоугольника:

```
int maxArea = 0;
for (int i = 0; i < n; ++i) {
    stack<int> indices;
    // ... (                                                    )
}
```

3. Вычисляется максимальная площадь, используя значения из стека после прохода всех столбцов строки.

Тест производительности



Оценка сложности: O($n * m$) где, m - длина образца.

Выводы

В данной лабораторной работе я познакомился с принципом решения задач, с помощью динамического программирования. Его основная идея заключается в том, чтобы разбить текущую задачу на элементарные подзадачи, которые мы уже умеем решать. Метод динамическо- го программирования позволяет решать большой класс задач за сложность, намного меньшую, чем наивную.