

Лабораторная работа № 7 по курсу дискретного анализа: Динамическое программирование

Выполнил студент группы 8О-207Б-17 МАИ *Ваньков Денис*.

Условие

При помощи метода динамического программирования разработать алгоритм решения задачи, определяемой своим вариантом; оценить время работы алгоритма и объем затраченной оперативной памяти. Перед выполнением необходимо применить метод динамического программирования.

Реализовать программу на *C* или *C++*, реализующую построенный алгоритм. Формат входных и выходных данных описан в варианте задания.

Вариант задания: 2. Пустой прямоугольник

Задан прямоугольник высотой n и шириной m , состоящий из нулей и единиц. Найдите в нем прямоугольник наибольшей площади, состоящий из одних нулей.

Входные данные: В первой строке заданы $1 \leq n \leq 500$ и $1 \leq m \leq 500$. В следующих n строках записаны по m символов 0 или 1 - элементы прямоугольника.

Выходные данные: Необходимо вывести одно число - максимальную площадь прямоугольника из одних нулей.

Метод решения

Динамическое программирование подразумевает разбиение основной задачи на несколько менее сложных подзадач. Этот метод применим для данной задачи, так как ее можно разбить на такие подзадачи.

Описание программы

Для хранения прямоугольника я использовал двумерный вектор *std::vector*, а также дополнительный вектор для хранения максимального количества нулей в столбце, для каждого из столбцов. Я искал максимальное количество нулей в столбце и передавал этот вектор, содержащий такие данные, в функцию, которая вычисляла максимальную площадь, начиная с самого маленького прямоугольника, и заканчивая самым большим. Это действие производилось для каждой из строк нашего прямоугольника. Сам подсчет площади основывался на предыдущих значениях, посчитанных алгоритмом на прошлом шаге, и хранящихся в структуре *std::stack*.

Дневник отладки

В процессе отладки удалось отследить ошибку внутри функции подсчета максимальной площади и исправить ее.

Выводы

Эта лабораторная работа не вызвала особых трудностей, так как здесь были необходимы знания простейших структур данных, а также умение разделять более сложную задачу на менее сложные. Моя программа использует $O(n*m)$ памяти, так как приходится хранить весь прямоугольник, и имеет $O(n*m)$ временную сложность, так как поиск максимального по площади идет по всем строкам и столбцам нашего исходного прямоугольника.