

Лабораторная работа № 7 по курсу дискретного анализа: динамическое программирование.

Выполнил студент группы М8О-208Б-18 МАИ *Коростелев Дмитрий Васильевич*.

Задание

Вариант №5

При помощи метода динамического программирования разработать алгоритм решения задачи, определяемой своим вариантом; оценить время выполнения алгоритма и объем затрачиваемой оперативной памяти. Перед выполнением задания необходимо обосновать применимость метода динамического программирования.

Разработать программу на языке С или С++, реализующую построенный алгоритм. Формат входных и выходных данных описан в варианте задания:

Задана матрица натуральных чисел A размерности $n * m$. Из текущей клетки можно перейти в любую из 3-х соседних, стоящих в строке с номером на единицу больше, при этом за каждый проход через клетку (i, j) взимается штраф $A_{i,j}$. Необходимо пройти из какой-нибудь клетки верхней строки до любой клетки нижней, набрав при проходе по клеткам минимальный штраф.

Формат входных данных

Первая строка входного файла содержит в себе пару чисел $2 \leq n \leq 1000$ и $2 \leq m \leq 1000$, затем следует n строк из m целых чисел.

Формат результата

Необходимо вывести в выходной файл на первой строке минимальный штраф, а на второй — последовательность координат из n ячеек, через которые пролегает маршрут с минимальным штрафом.

Метод решения

Для решения данного задания воспользуемся методом динамического программирования, и разобьем задачу на более простые задачи. Так требуется найти кратчайшие через поле размерности $n*m$ при этом на каждом шаге нужно спускаться на одну строку ниже. Всего таких строк N и размер каждой — M . Наивный алгоритм решения данной задачи — полный перебор всех возможных путей. Таким образом на каждом шаге выбираем 1 из 3 (2-ух если находимся на краю поля) переходов, повторяем это действие, запоминаем все длины всех путей, выбираем из получившихся путей путь с самым маленьким штрафом, восстанавливаем его и получаем результат. Асимптотика такого решения $O(m * 3^n)$.

Теперь используя принципы динамического программирования, разобьём задачу на несколько подзадач. Заданное поле представляет из себя набор из n слоев, где на каждом из них мы должны выбирать путь с минимальным штрафом. И тогда для решения такой подзадачи требуется, чтобы минимальный штраф на более низком уровне уже был найден – получаем индукцию: мы знаем как найти минимальный путь в текущем слое (смотри на след. слой, выбираем ячейку с минимальным штрафом и прибавляем к ней штраф текущей ячейки) и штрафы для последнего слоя – штрафы самой последней строки исходного массива.

После прохода по матрице в верхней строчке выбираем наименьшее значение и восстанавливаем путь.

Асимптотика решения

Выполняем n подзадач каждая из которых выполняется за $O(m)$ – итоговое время работы – $O(n * m)$

Память, расходуемая при решении также равна $O(n * m)$ так как хранится сама матрица штрафов.

Отладка и проверка программы.

№	Название ошибки	Причина возникновения ошибки
1-2	Неправильный формат вывода	Отсутствовали разделители
3-8	3 Неправильный ответ	Переполнение типа int

Недочеты

Создание второй матрицы, которая копирует исходную (исходная матрица тоже сохраняется).

Вывод

Не сложно убедиться, что для решения некоторых задач наивное, прямое решение зачастую может оказаться не самым выгодным по памяти или быстрым по времени. Каждую задачу следует тщательно обдумывать и применять к ней особые методы, оптимизирующие решение задачи.