

Лабораторная работа № 8 по курсу дискретного анализа: динамическое программирование

Выполнил студент группы М08-312Б МАИ *Лобанов Олег*.

Условие

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.

Входные данные: 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.

Выходные данные: 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.

Вариант 5: Обход матрицы.

Метод решения

Оптимальная подструктура в динамическом программировании означает, что оптимальное решение подзадач меньшего размера может быть использовано для решения исходной задачи. В общем случае мы можем рашить задачу, в которой присутствует оптимальная подструктура, проделывая следующие три шага:

- Разбиение задачи на подзадачи меньшего размера.
- Нахождение оптимального решения подзадач рекурсивно, проделывая такой же трехшаговый алгоритм.
- Использование полученного решения подзадач для конструирования решения исходной задачи.

Подзадачи решаются делением их на подзадачи еще меньшего размера и т.д., пока не приходят к тривиальному случаю задачи, решаемой за константное время.

Для решения задачи будем использовать динамическое программирование, так как в этой задаче можно последовательно посчитать ответ для всех ячеек массива, на основании результатов предыдущих ячеек. В противном случае, мы бы получили перебор значений с огромной сложностью, не меньше чем $O(n^3)$, когда как динамическое программирование решает эту проблему за $O(n^2)$.

Описание программы

Minimum_of_3 - функция отвечающая за сравнение 3-х элементов матрицы и выбор минимальной из них.

Сначала записываются данные в одномерный массив. Далее, идя снизу вверх, обновляю каждый элемент массива В путем добавления текущего элемента массива А к минимальному из трех возможных вариантов движения. В Конце результат минимальный суммы и полный путь выводятся на стандартный вывод.

Дневник отладки

В первой попытке выдался неверный ответ на первом тесте. Ошибка достаточно быстро нашлась - я случайно к позиции элемента массива В приплюсовывал элемент массива А.

Во второй попытке выдался неверный ответ на восьмом тесте. Ошибка искалась немалое время. Оказалось, что я забыл правильно ввести начально значение k и сравнивал одинаковые позиции.

На третьей попытке все "окнулось".

Выводы

После выполнения восьмой лабораторной работы я научился использовать динамическое программирование. Это одна из самых простых лабораторных работ. Но из-за моей невнимательности происходили нелепые ошибки, которые не должны были допускаться. В процессе решения я понял, что придумать рекурсивную формулу не такая уж и тривиальная задача.