

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

Выполнил студент группы М8О-307-20 МАИ *Михеева Кристина*.

Условие

1. 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.

Вариант №3 (Количество чисел):

Задано целое число n . Необходимо найти количество натуральных (без нуля) чисел, которые меньше n по значению и меньше n лексикографически (если сравнивать два числа как строки), а так же делятся на m без остатка.

Метод решения

Для более оптимального решения данной задачи был использован метод динамического программирования, который позволяет разбить исходную задачу на более мелкие подзадачи и начать решать их при этом компоновка подзадач должна вызывать меньше времени.

В данной задаче следует разбить решение на сумму решений на интервалах от 10^i до i , где i это какая-то подстрока исходного числа m и считаем на них количество таких чисел, которые лексикографически меньше n и n при делении на них не дает остатка.

Данным решением мы оптимизировали задачу, что вместо $O(n)$ получится $O(k)$, где k - количество цифр в данном числе n .

Описание программы

Программа состоит из одного файла lab7.cpp. На вход мы принимаем два исходных числа n , m , и сразу задаем нижнюю и верхнюю границу. Заводим цикл, в котором мы считаем минимальное число, которое должно делиться на m . Еще в этом цикле ищем также и максимальное число, которое тоже должно делиться на m . Затем считается ответ на каждом из интервалов.

Дневник отладки

Ошибок не найдено.

Тест производительности

Наивный алгоритм:

Тесты: 42 3; 1000 10; 100000 100

Время выполнения: 1.307; 4.199; 8.071;

Динамическое программирование:

Тесты: 42 3; 1000 10; 100000 100;

Время выполнения: 0.675; 1.787; 3.098;

Видим, что методом динамическим программированием, решение получилось значительно быстрее, чем наивным способом.

Выводы

В данной лабораторной работе №7 было предложено решить некоторую задачу с помощью динамического программирования, которая позволила оптимизировать решение и заставить алгоритм работать быстрее. Все это благодаря разбиению основной задачи на более мелкие и простые подзадачи, где сумма их времени выполнения должна быть меньше, чем основная задача. Я считаю, что это полезное знание, которое может пригодиться в дальнейшей работе для ускорения решений задач.