Лабораторная работа № 7 по курсу дискретного анализа: Динамическое программирование

Выполнил студент группы М8О-308Б-20 МАИ *Попов Матвей*.

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.

2. **Вариант 1: Хитрый рюкзак.** У вас есть рюкзак, вместимостью m, а также n предметов, у каждого из которых есть вес w_i и стоимость c_i. Необходимо выбрать такое подмножество I из них, чтобы

   - $\sum\limits_{i \in I} w_i \leq m$
   - $(\sum\limits_{i \in I} c_i) * |I|$ является максимальной из всех возможных

   $|I|$ — мощность множества I.

Метод решения

Для решения воспользуемся методом динамического программирования, в котором большая задача разбивается на множество маленьких подзадач, решения каждой из которых может быть использовано несколько раз, что сильно увеличивает производительность алгоритма. Вспомним решение изветсной задачи о рюкзаке: составим таблицу из n строк и m столбцов и заполним каждую ячейку по определённому правилу, причём в заполнении каждой ячейки будут участвовать уже заполненные ячейки, в итоге в правой нижней ячейке нашей таблицы будет лежать решение задачи, то есть максимальная стоимость, которую можно унести в рюкзаке. Задача из моего варианта отличается от оригинальной условием заполнения таблицы, а также тем, что нужно хранить вспомогательную таблицу для восстановления последовательности предметов.

Описание программы

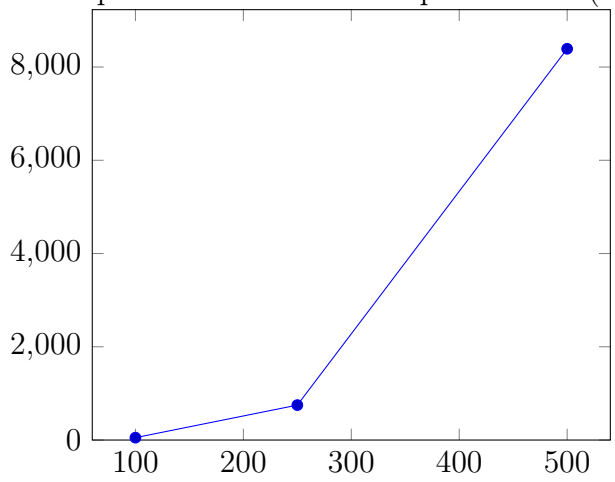
Программа состоит из одного файла.

Дневник отладки

1. Было несколько неудачных попыток, все они связаны с неправильным выбором типа переменной для хранения стоимости предметов.

Тест производительности

Ниже приведен тест времени работы алгоритма. По оси X — количество предметов, по оси Y — время выполнения алгоритма в мс (меньше — лучше).



Кол-во предметов	Время (в мс)
100	51
250	749
500	8391

Тесты подтвердили временную сложность алгоритма — $O(n^2m)$, что гораздо лучше простого перебора за $O(2^n)$

Недочёты

В процессе выполнения работы не использовалось дополнительных классов или функций, из-за чего программисту, незнакомому с алгоритмом, будет очень сложно разбираться в коде.

Выводы

Проделав лабораторную работу, познакомился с новым подходом решения алгоритмических задач — динамическим программированием, а также впервые за всё время использовал класс std::bitset в своей программе.