

Лабораторная работа №7 по курсу дискретного анализа: динамическое программирование

Выполнил студент группы М8О-307Б-21 МАИ *Кажекин Денис*

Условие

Необходимо было решить стандартную задачу на динамическое программирование «рюкзак», в которой нужно собрать максимально дорогой рюкзак (стоимость предметов, помноженная на их количество, должна быть максимальной). Вариант задания – 1.

Метод решения

Задача была решена с помощью подхода динамического программирования. То есть: ответ был найден с помощью разбиения задачи на несколько более простых, с помощью решения которых, можно находить ответ на текущем шаге. Еще раз хочу подметить, что классическая задача была усложнена дополнительным условием об умножении стоимости рюкзака на мощность множества предметов в рюкзаке, что немного усложнило процесс. Помимо матрицы, которая использовалась для нахождения ответа, понадобилась еще одна для восстановления ответа.

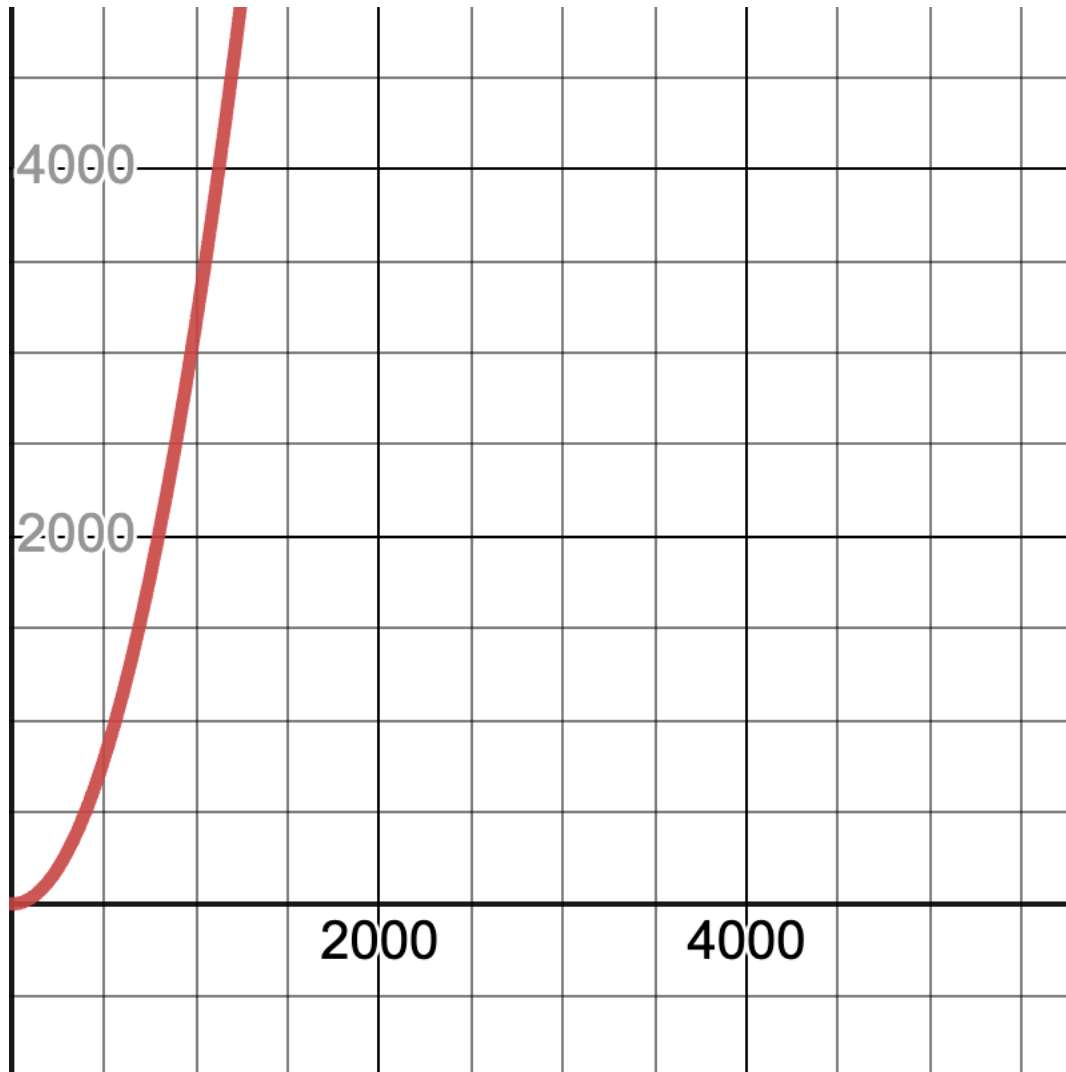
Описание программы

Программа состоит из одного файла `knapsack.cpp`

Дневник отладки

Реализовав стандартный алгоритм для решения задачи о рюкзаке заметил, что условие немного отличается, после чего пришлось добавить еще одну размерность в матрицу динамического программирования, чтобы получить верный ответ.

Тест производительности



В ходе тестов было обнаружено, что программа выдает сложность $O(n^2 * m)$. Не лучшее, что хотелось бы получить, но для такой задачи очень даже эффективно по сравнению с полным перебором за $O(2^n)$

Недочёты

Слишком нагруженный код, без декомпозиции.

Выводы

В ходе выполнения лабораторной работы я познакомился с парадигмой решения задач «Динамическое программирование». Это дало мне бесценный опыт и я надеюсь, что это поможет мне найти решение в будущих задачах наиболее эффективным способом