

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

Выполнил студент группы М8О-307Б-20 Мерц Савелий Павлович

## Условие

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

## Вариант:

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:

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

$|I|$  — мощность множества  $I$ .

## Метод решения:

Основная идея динамического программирования заключается в том, что сложная задача разбивается на более простые и решение сложной задачи состоит из решений более простых задач. Задача о рюкзаке является известной NP-полной задачей, которая при некоторых ограничениях решается за полиномиальное время с помощью метода динамического программирования.

Для моего варианта задания  $dp_{i,j,k}$  — максимальная стоимость  $j$  вещей из первых  $i$ , таких, что их суммарный вес не превышает  $k$ . То есть алгоритм будет перебирать количество предметов, которые будут в рюкзаке.

Пусть существует оптимальное решение в  $dp_{i,j,kw,j+1}$ , тогда  $dp_{i+1,j+1,k} = \max(dp_{i,j,kw,j+1} + c_{j+1}, dp_{i+1,j,k})$ . В рекуррентной формуле рассматривается два варианта: взять вещь  $j + 1$  или нет.

Такое решение имеет  $n^2 * m$  состояний, в каждое можно перейти из двух других. Временная сложность алгоритма  $O(n^2 * m)$ .

В памяти будет храниться только  $dp_i$ ,  $dp_{i+1}$  и битовые множества предметов, которые оптимальны для решения подзадачи. Пространственная сложность такого подхода  $O(nm)$ .

### **Описание программы:**

В единственном файле программы реализовано решение.

### **Дневник отладки:**

Сначала пытался жадно решить, что не дало нужного результата, после чего пришлось долго думать над решением, реализация не вызвала осложнений.

### **Тест производительности:**

Кол-во предметов	Время
10	3
100	51
250	739
500	8393

Тесты подтвердили временную сложность алгоритма —  $O(n * m)$

### **Выводы**

В ходе выполнения лабораторной работы я изучил классические задачи динамического программирования и методы их решения. А также реализовал алгоритм для своего варианта задания. Хотя жадные алгоритмы и точны, но не достаточно для данной задачи.