

## Лабораторная работа №7 по курсу «Дискретный анализ» по теме «Динамическое программирование»

Выполнил студент группы М8О-308Б-22 Иванов Андрей

### Условие: (вариант 1 – хитрый рюкзак)

У вас есть рюкзак, вместимостью  $m$ , а так же  $n$  предметов, у каждого из которых есть вес  $w_i$  и стоимость  $c_i$ . Необходимо выбрать такое подмножество  $I$  из них, чтобы:

$$\cdot \sum_{i \in I} w_i \leq m$$

$$\cdot \left( \sum_{i \in I} c_i \right) * |I|$$
 является максимальной из всех возможных.  
 $|I|$  – мощность множества  $I$ .

Формат ввода: в первой строке заданы  $1 \leq n \leq 100$  и  $1 \leq m \leq 5000$ . В последующих  $n$  строках через пробел заданы параметры предметов:  $w_i$  и  $c_i$ .

Формат вывода: в первой строке необходимо вывести одно число – максимальное

значение  $\left( \sum_{i \in I} c_i \right) * |I|$ , а на второй – индексы предметов, входящих в ответ.

### Описание алгоритма

Алгоритмы, предназначенные для решения задач оптимизации, обычно представляют собой последовательность шагов, на каждом из которых предоставляется некоторое множество выборов. В динамическом программировании исходная задача делится на связанные подзадачи и, помня результаты всех решенных подзадач на прошлом шаге, определяем наилучшее решение на текущем.

Данная задача не является классической задачей о рюкзаке, где надо максимизировать суммарную стоимость. Вместо этого надо максимизировать число, равное этой суммарной стоимости умноженной на мощность множества взятых предметов.

### Описание программы

Будем хранить элементы в матрице размерами (вместимость рюкзака)  $X$  (индекс предмета)  $X$  (мощность подмножества). Соответственно, обойдем этот трехмерный тензор тремя циклами, и на каждом шаге попытаемся положить вещь в рюкзак. Если мощность выбранного подмножества превосходит количество оставшихся элементов, то такое количество элементов выбрать невозможно, также как и невозможно засунуть в рюкзак предмет, превосходящий вместимость рюкзака на определенном шаге. Иначе смотрим, что выгоднее: брать или не брать предмет. Это решение запоминаем, чтобы использовать на следующем шаге. Для оптимизации по памяти будем хранить в векторе только текущее и предыдущее

значения, так как на каждом шаге ДП проверяет именно предыдущее.

После этого проходим по всем возможным мощностям подмножеств и выбираем максимально возможное значение произведения мощности и стоимости. После этого выводим решение, используя указатели на предыдущий объект внутри структуры.

### **Сложность**

Сложность равна  $O(n * n * m)$ , так как мы два раза перебираем  $n$  шагов (количества и мощности) и 1 раз перебираем вместимости  $m$ .

### **Выводы**

Благодаря лабораторной работе была изучена необычная задача, связанная с вместимостью рюкзака. Ее решение было осуществлено с помощью динамического программирования – способа решения сложных задач путем разбиения их на более простые подзадачи и последующего комбинирования их решений.