

# Лабораторная работа 6

## Жадная стратегия и динамическое программирование

(25 баллов)

Выполните самостоятельно следующие задания и оформите отчет.

Требования по отчету:

Наличие титульного листа. Размер страницы должен соответствовать формату А4 (210x297), размеры полей: левое – 30 мм, правое – 10 мм, верхнее – 15 мм, нижнее – 20 мм. Шрифт Times new Roman, размер 14 pt полуторный междустрочный интервал. Выравнивание текста – по ширине, красная строка – 1,25 см, отступ слева и справа – 0 мм.

### № 1

(2 балла)

Предположим, что вы собираетесь в турпоход. У вас есть рюкзак, емкость которого составляет 6 фунтов, и список предметов, которые вы можете положить с собой в рюкзак. У каждого предмета есть вес и ценность. Чем выше ценность, тем важнее для вас предмет.

Нарисуйте и вручную заполните таблицу, которая даст оптимальный набор предметов для похода и общую ценность получившегося рюкзака. Ориентируйтесь на алгоритм динамического программирования, описанный в лекции.

Список предметов возьмите согласно вашему варианту.

#### Вариант 1.

1. вода, 3 фунта, 5;
2. книга, 1 фунт, 3;
3. еда, 2 фунта, 6;
4. куртка, 2 фунта, 5;
5. камера, 1 фунт, 6

#### Вариант 2.

1. вода, 1 фунт, 10;
2. книга, 2 фунта, 3;
3. еда, 2 фунта, 9;
4. куртка, 1 фунт, 5;
5. камера, 4 фунта, 6

#### Вариант 3.

1. вода, 2 фунта, 8;
2. книга, 2 фунта, 5;
3. еда, 3 фунта, 7;
4. куртка, 1 фунт, 3;

5. камера, 2 фунта, 4

Вариант 4.

1. вода, 2 фунта, 6;
2. книга, 1 фунт, 3;
3. еда, 4 фунта, 9;
4. куртка, 1 фунт, 5;
5. камера, 3 фунта, 6

Вариант 5.

6. вода, 1 фунт, 8;
7. книга, 1 фунт, 8;
8. еда, 3 фунта, 9;
9. куртка, 2 фунта, 5;
10. камера, 2 фунта, 6

**№ 2**

**(3 балла)**

Напишите программу, которая моделирует задачу о рюкзаке. Пользователь вводит объем рюкзака и список предметов с указанием их веса и ценности. Вам необходимо подобрать набор предметов максимальной ценности, которые поместятся в рюкзак. Реализуйте три варианта решения этой задачи:

1. Полный перебор
2. Жадный алгоритм
3. Динамическое программирование

Укажите вычислительная сложность (в O-нотации) для каждого алгоритма. Постройте графики возрастания времени выполнения алгоритмов при увеличении входных данных.

Приведите примеры входных данных, когда жадный алгоритм дает оптимальное решение и когда дает неоптимальное решение.

**№ 3**

**(2 балла)**

Нарисуйте и вручную заполните таблицу для вычисления самой длинной общей подстроки между строками blue и clue. Ориентируйтесь на алгоритм динамического программирования, описанный в лекции.

**№ 4**

**(3 балла)**

Напишите программу, реализующую алгоритм нахождения самой длинной общей подстроки из лекции. Пользователь вводит слово с ошибкой и далее список похожих слов. Ваша программа должна вывести самое похожее слово согласно данному алгоритму.

## № 5

(3 балла)

Напишите программу, реализующую алгоритм нахождения самой длинной общей подпоследовательности из лекции. Пользователь вводит слово с ошибкой и далее список похожих слов. Ваша программа должна вывести самое похожее слово согласно данному алгоритму.

## № 6

(4 балла)

Имеется  $n$  предметов различных размеров (от 0.1 до 1.0). Есть неограниченное количество одинаковых ящиков вместимостью 1.0. Необходимо разложить все  $n$  предметов в минимальное количество ящиков.

Напишите программу, реализующую 4 жадных стратегии раскладки предметов по ящикам:

1. Первый подходящий ящик. Ящики просматриваются поочередно пока не найдется ящик, в котором достаточно свободного места для упаковки очередного предмета. Если такого ящика нет, то предмет кладется в новый ящик.  
Например, упаковка предметов размерами (0.5, 0.7, 0.3, 0.9, 0.6, 0.8, 0.1, 0.4, 0.2, 0.5) пройдет следующим образом: первый ящик — [0.5, 0.3, 0.1], второй — [0.7, 0.2], третий — [0.9], четвертый — [0.6, 0.4], пятый — [0.8], шестой — [0.5]. Т.е. ответом будет 6 ящиков.
2. Наиболее подходящий ящик. Выбирается ящик, в котором останется как можно меньше места после помещения туда очередного предмета. Укладка в новый ящик применяется только в том случае, если очередной предмет не помещается ни в какой из имеющихся ящиков.  
Например, упаковка предметов размерами (0.5, 0.7, 0.3, 0.9, 0.6, 0.8, 0.1, 0.4, 0.2, 0.5) пройдет следующим образом: первый ящик — [0.5, 0.5], второй — [0.7, 0.3], третий — [0.9, 0.1], четвертый — [0.6, 0.4], пятый — [0.8, 0.2]. Т.е. ответом будет 5 ящиков.
3. Следующий подходящий ящик. Мы продолжаем укладку каждого ящика до тех пор, пока очередной предмет в него помещается. Если предмет не влезает в ящик, то мы берем новый ящик, а к ранее уложенным ящикам уже не возвращаемся.  
Например, упаковка предметов размерами (0.5, 0.7, 0.3, 0.9, 0.6, 0.8, 0.1, 0.4, 0.2, 0.5) пройдет следующим образом: первый ящик — [0.5], второй — [0.7, 0.3], третий — [0.9], четвертый — [0.6], пятый — [0.8, 0.1], шестой — [0.4, 0.2], седьмой — [0.5]. Т.е. ответом будет 7 ящиков.
4. Наименее подходящий ящик. Выбирается ящик, в котором останется как можно больше места после помещения туда очередного предмета. Укладка в новый ящик применяется только в том случае, если очередной предмет не помещается ни в какой из имеющихся ящиков.  
Например, упаковка предметов размерами (0.5, 0.7, 0.3, 0.9, 0.6, 0.8, 0.1, 0.4, 0.2, 0.5) пройдет следующим образом: первый ящик — [0.5, 0.3], второй — [0.7], третий —

[0.9], четвертый — [0.6, 0.1], пятый — [0.8], шестой — [0.4, 0.2], седьмой — [0.5].  
Т.е. ответом будет 7 ящиков.

Сгенерируйте случайные наборы предметов (размером от 0.1 до 0.9) в количестве 50, 100, 200 и 500. Осуществите на каждом из наборов проверку всех четырех стратегий и посмотрите, какая из них приводит к меньшему числу ящиков.

В отчете приведите ваши результаты в виде таблицы.

## № 7

(4 балла)

Есть купюры и монеты номиналами: 1, 3, 4, 10, 50, 100. В банкомате неограниченное количество купюр и монет каждого номинала. Мы хотим снять со счета  $n$  рублей. Нужно вывести на экран минимальный набор купюр и монет, который может выдать банкомат, чтобы сумма получилась ровно  $n$ .

1. Напишите жадный алгоритм решения задачи. Подсказка: выбирайте купюры и монеты по убыванию номинала.
2. Используйте динамическое программирование, чтобы рассчитать минимальное количество требуемых купюр и монет.
3. Приведите пример входных данных, когда жадный алгоритм выдает не оптимальное решение.

## № 8

(4 балла)

Напишите программу, моделирующую работу клиентов с банкоматом. В банкомате есть купюры и монеты номиналами: 1, 3, 4, 10, 50, 100, но, в отличие от предыдущей задачи, их количество конечно.

В начале работы программы вы задаете:

- количество клиентов
- для каждого клиента количество рублей, которые он хочет снять (считаем, что эта сумма всегда корректна и имеется на счету клиента)
- изначальное количество купюр и монет в банкомате

Далее вы по очереди обрабатываете запросы клиентов. Для каждого клиента выведите на экран минимальный набор купюр и монет для его суммы или сообщение о невозможности предоставить данную суммы из-за нехватки купюр и монет. Если минимальный набор найден успешно, то вычтите его из запасов банкомата и выведите на экран получившейся остаток.

Проведите несколько запусков программы с разными начальными данными и различными исходами работы с банкоматом.