**Практическая работа №4**

**Оценка сложности эвристических алгоритмов**

**Цель работы:**

Научиться оценивать сложность эвристических алгоритмов.

**Задание:**

Оценить сложность эвристического алгоритма.

Жадный алгоритм

Задача:

Монетная система некоторого государства состоит из монет достоинством  ​, причём каждый следующий номинал делится на предыдущий. Требуется выдать сумму с наименьшим возможным количеством монет.

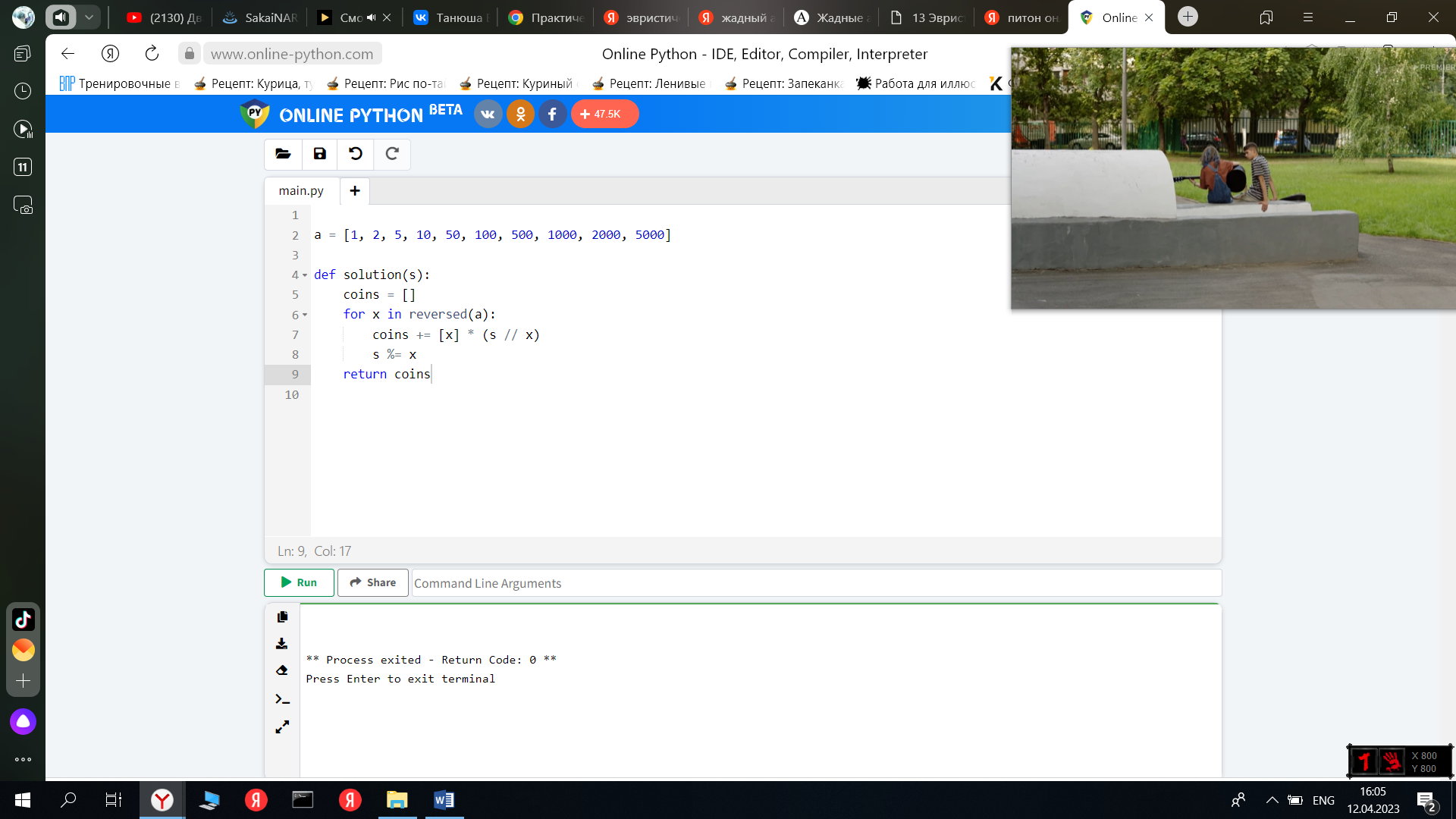


Рис.1.Программный код

Заметим, что без предположения, что номиналы монет делятся друг на друга, алгоритм не всегда выводит оптимальное решение, и в этом случае задачу остается решать динамическим программированием. Например, сумму в 24 рубля монетами в 1, 5 и 7 рублей жадный алгоритм разменивает как (7×3+1×3), в то время как оптимальным будет (7×2+5×2).

Оценим сложность данного алгоритма

Докажем корректность жадного алгоритма от противного: пусть максимальная монета имеет достоинство , но в оптимальном ответе её нет.

Пусть в оптимальном ответе монета с номиналом встретилась раз:

Если , то монет можно заменить на одну монету достоинства , а значит это не оптимальный ответ. Следовательно,  .

Теперь посчитаем, какой может быть максимальная сумма всех достоинств всех монет в оптимальном ответе, если мы не брали максимальные монеты:

Отсюда делаем вывод, что если , то в оптимальный ответ всегда придется взять максимальную монету размера , потому что все меньшие монеты просто не смогут оптимальном ответе давать так много. Аналогичным размышлением про наборы из меньших монет получаем такой же результат для всех , откуда следует корректность жадного алгоритма.

Также этот алгоритм работает не только для делящихся друг на друга номинациях, но и для некоторых других. Такие монетные системы, где жадный алгоритм работает, называют каноническими O(1).

**Контрольные вопросы:**

1. Что такое эвристический алгоритм?

Это алгоритм решения задачи, правильность которого для всех возможных случаев не доказано. Но про который известно, что он даёт достаточно хорошее решение в большинстве случаев.

1. Что такое формулы оценки сложности алгоритмов?

Сложность алгоритмов обычно оценивают по времени выполнения или по используемой памяти. В обоих случаях сложность зависит от размеров входных данных: массив из 100 элементов будет обработан быстрее, чем аналогичный из 1000. При этом точное время мало кого интересует: оно зависит от процессора, типа данных, языка программирования и множества других параметров. Важна лишь асимптотическая сложность, т. е. сложность при стремлении размера входных данных к бесконечности.

**Вывод:**

Научиться оценивать сложность эвристических алгоритмов.