**Министерство науки и высшего образования Российской Федерации**ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ АВТОНОМНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ  
**НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО**

**Факультет безопасности информационных технологий**

**Дисциплина:**

«Алгоритмы и структуры данных»

**ОТЧЕТ ПО ЛАБОРАТОРНОЙ РАБОТЕ № 1**

«Сортировка методом Монте-Карло»

**Выполнил:**

Гачко Г. Д., студент группы N3246

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

**Проверил:**

Ерофеев С. А.

*\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_*

(подпись)

Санкт-Петербург

2024 г.

Содержание

[Введение 4](#_Toc159025670)

[1 Описание функционала программы 5](#_Toc159025671)

[2 Блок-схема 6](#_Toc159025672)

[3 Код программы 7](#_Toc159025673)

[4 Результаты тестирования 8](#_Toc159025674)

[Заключение 10](#_Toc159025675)

Введение

Задача работы – разработать программу сортировки массива чисел из файла методом Монте-Карло.

Для выполнения задачи требуется реализовать ввод чисел из файла, выполнение сортировки и вывод чисел в файл.

Для реализации был выбран язык программирования Python версии 3.10.4.

# Описание функционала программы

Программа принимает на вход из stdin режим запуска – ручное управление (M) или запуск тестов (T). Программа обращается к файлу с входными данными по жестко указанному в коде относительному пути к файлу. Это решение было принято из соображений простоты использования программы. Далее будут рассмотрены два режима работы программы.

**M**anual:

В этом режиме программы в бесконечном цикле выполняет сортировку файла numbers\_valid.txt. Как только сортировка завершена, программа выводит в stdout результирующий массив и ожидает подтверждения следующей сортировки путем нажатия клавиши Enter. Между сортировками входной файл numbers\_valid.txt может быть изменен пользователем для проверки функционирования программы при различных входных данных.

**T**est:

В этом режиме программа не принимает пользовательский ввод, а запускается на наборе заранее написанных тестов. Встроенные в программу тесты включают проверку следующих случаев:

1. Во входном файле содержатся данные, не приводимые к типу float (буквы, знаки препинания, пробельные символы и т.п.)
2. Во входном файле содержатся числа, превышающие размер типа данных float для языка Python 3.11
3. Во входном файле содержатся корректные данные: отрицательные и положительные дробные и целые числа, ноль.

В обоих режимах работы алгоритм программы выглядит следующим образом. При получении данных из файла программа проводит проверку данных на приводимость к типу данных float, и завершается с ошибкой, если данные из файла невозможно преобразовать к типу float. Также проводится проверка на переполнение типа данных float.

После запуска программа считывает массив чисел из входного файла и закрывает его. Далее проводится случайная перестановка чисел в массиве до тех пор, пока массив не будет отсортирован. После выполнения сортировки программа записывает отсортированный массив в файл, отличный от файла ввода, и выводит получившийся массив в stdout.

# Используемые переменные

filenameForTest1, filenameForTest2, filenameForTest3, filenameOutput – имена файлов для работы программы (тип string, диапазон ограничен памятью устройства)

numbers – массив сортируемых чисел (тип float List, максимальная длина – 9223372036854775807 элементов)

content – переменная для промежуточного хранения данных во время чтения из файла (тип string, диапазон ограничен памятью устройства)

# Блок-схема

Изображение выглядит как текст, диаграмма, линия, зарисовка

Автоматически созданное описание

# Код программы

import random

# Файлы с тестовыми данными:

# Тест 1

# String filenameForTest1 - относительный путь к файлу с входными данными для теста 1

filenameForTest1 = 'numbers\_with\_string.txt'

# Тест 2

# String filenameForTest2 - относительный путь к файлу с входными данными для теста 2

filenameForTest2 = 'numbers\_float\_overflow.txt'

# Тест 3

# String filenameForTest3 - относительный путь к файлу с входными данными для теста 3

filenameForTest3 = 'numbers\_valid.txt'

# Файл вывода

# String filenameOutput - относительный путь к файлу с выходными данными

filenameOutput = 'sorted\_numbers\_output.txt'

def sortMonteCarlo(input\_filename = filenameForTest3, output\_filename = filenameOutput, testing\_mode = False):

    # Float List numbers - Массив, в котором содержатся сортируемые числа в ходе работы программы

    numbers = []

    # Ввод из файла

    with open(input\_filename, 'r') as file:

        # String List content - относительный путь к файлу с входными данными

        content = file.read().splitlines()

    # Конвертация полученного массива строк в массив чисел типа float

    for str\_number in content:

        try:

            numbers.append(float(str\_number))

        # Обработка случая, когда в файле предоставлены не float

        except ValueError:

            print("ОШИБКА: Файл содержит данные, не приводимые к типу float")

            return

        # Обработка случая, когда в файле предоставлены числа, большие float

        if (float(str\_number) == float(str\_number) + 1): # raise OverflowError() - имеется в виду обработка этого исключения

            print("ОШИБКА: Файл содержит числа, превышающие размер типа данных float")

            return

    # Сортировка

    # Цикл проверки массива на отсортированность

    while not isSorted(numbers):

        # "Сортировка" массива методом Монте-Карло

        random.shuffle(numbers)

    # Запись в файл

    with open(output\_filename, 'w') as file:

        for i in range(len(numbers)):

            file.write(str(numbers[i]) + '\n')

    print(numbers) # НЕ отладочный вывод результата работы в stdout

    return

# Небольшая функция, проверяющая сортировку массива за O(N)

def isSorted(array):

    for i in range(len(array) - 1):

        if array[i] < array[i + 1]:

            return False

    return True

mode = input("Какой режим запуска? (run [T]ests / run [M]anually)")

if mode == 'M':

    while True:

        # функция сортировки массива будет запускаться заново при каждом нажатии Enter,

        # между итерациями можно менять файл с входными данными

        sortMonteCarlo()

        input("")

elif mode == 'T':

    # Три теста

    # Тест 1: Случай со строковыми данными в файле

    print("\nТест 1: Не-float данныe в файле")

    print("Содержимое файла " + filenameForTest1)

    with open(filenameForTest1, 'r') as file:

        print(file.read())

    print("\nВывод программы:")

    sortMonteCarlo(input\_filename = filenameForTest1, testing\_mode = True)

    # Тест 2: Случай с переполнением типа данных float

    print("\nТест 2: float в файле выходит за размер типа данных")

    print("Содержимое файла " + filenameForTest2)

    with open(filenameForTest2, 'r') as file:

        print(file.read())

    print("\nВывод программы:")

    sortMonteCarlo(input\_filename = filenameForTest2, testing\_mode = True)

    # Тест 3: Случай с валидным файлом

    print("\nТест 3: Валидный файл ввода: дробные числа, положительные числа, отрицательные числа, ноль.")

    print("Содержимое файла " + filenameForTest3)

    with open(filenameForTest3, 'r') as file:

        print(file.read())

    print("\nВывод программы:")

    sortMonteCarlo(input\_filename = filenameForTest3, testing\_mode = True)

else:

    print("Вы ввели некорректный режим запуска. Введите T (для запуска тестов) или M (для запуска в ручном режиме)")

# Результаты тестирования

Были проведены тесты программы с различными входными данными. Все реализованные случаи ввода корректно обрабатываются и отображаются. Тестирование производилось с помощью интерпретатора Python 3.10.4 и среды разработки [ms-python.python](https://marketplace.visualstudio.com/items?itemName=ms-python.python) версии 2024.3.10791010

Изображение выглядит как текст, снимок экрана, Шрифт, документ

Автоматически созданное описание

# Заключение

В ходе выполнения данной лабораторной работы была достигнута поставленная задача. А именно, был составлен и реализован алгоритм по сортировке массива числе методом Монте-Карло со вводом из файла.

В ходе решения данной задачи была реализована функция «sortMonteCarlo», принимающая на вход файл с числами, подлежащими сортировке, а также сортирующая числа в этом файле, и выводящая результат в выходной файл.

Также была реализована функция «isSorted» для проверки того, отсортирован массив или нет.

Также были написаны тестовые файлы с различными (валидными и не валидными) входными данными, и написаны тесты программы в автоматическом режиме с их использованием.

Было выполнено тестирование программы на языке Python версии 3.10.4 с помощью среды разработки [ms-python.python](https://marketplace.visualstudio.com/items?itemName=ms-python.python) версии 2024.3.10791010, в ходе которого все реализованные выводы оказались корректно работающими.