

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

ОТЧЕТ ПО УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКЕ
(2022/2023 учебный год)

Шатравин Дмитрий Андреевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А.

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Зинкин С.А.

(должность, ученая степень, ученое звание)

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

Утвержден на заседании кафедры

«Вычислительная техника» _____

" ____ " _____ 20 ____ г.

Заведующий кафедрой

_____ М.А. Митрохин

**ИНДИВИДУАЛЬНЫЙ ПЛАН ПРОХОЖДЕНИЯ УЧЕБНОЙ
(ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ**

(2022/2023 учебный год)

Шатравин Дмитрий Андреевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Заведующий кафедрой д.т.н., профессор, Митрохин М.А. _____

(должность, ученая степень, ученое звание, Ф.И.О.)

Руководитель практики д.т.н., профессор, Зинкин С.А. _____

(должность, ученая степень, ученое звание)

№ п/п	Планируемая форма работы во время практики	Количество часов	Календарные сроки проведения работы	Подпись руководителя практики от вуза
1	Выбор темы и разработка индивидуального плана проведения работ	2	29.06.2023 - 29.06.2023	
2	Подбор и изучение материала по теме работы	15	30.06.2023 – 02.07.23	
3	Разработка алгоритма	43	02.07.23 – 06.07.23	
4	Описание алгоритма и программы	18	6.07.23 – 08.07.23	
5	Тестирование	5	08.07.23 – 08.07.23	
6	Получение и анализ результатов	10	08.07.23 – 10.07.23	
7	Оформление отчёта	15	10.07.23 – 12.07.2023	
	Общий объём часов	108		

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЧЁТ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2022/2023 учебный год)

Шатравин Дмитрий Андреевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения _____ 1 _____ семестр _____ 2 _____

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

Шатравин Д.А. выполнял практическое задание «Сортировка выбором». На первоначальном этапе был изучен и проанализирован алгоритм сортировки выбором, был выбран метод решения и язык программирования Python, на котором была написана программа сортировки массива методом выбора. Написал функции по работе с файлами, протестировал программу. Оформил отчёт.

Бакалавр Шатравин Д.А. _____ "___" _____ 2023 г.

Руководитель Зинкин С.А. _____ "___" _____ 2023 г.
практики

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ
ФЕДЕРАЦИИ
ПЕНЗЕНСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ
ПОЛИТЕХНИЧЕСКИЙ ИНСТИТУТ
ФАКУЛЬТЕТ ВЫЧИСЛИТЕЛЬНОЙ ТЕХНИКИ

ОТЗЫВ

О ПРОХОЖДЕНИИ УЧЕБНОЙ (ОЗНАКОМИТЕЛЬНОЙ) ПРАКТИКИ

(2022/2023 учебный год)

Шатравин Дмитрий Андреевич

Направление подготовки 09.03.01 «Информатика и вычислительная техника»

Наименование профиля подготовки «Программное обеспечение средств
вычислительной техники и автоматизированных систем»

Форма обучения – очная Срок обучения в соответствии с ФГОС – 4 года

Год обучения 1 семестр 2

Период прохождения практики с 29.06.2023 по 12.07.2023

Кафедра «Вычислительная техника»

В процессе выполнения практики Шатравин Д.А. решал следующие задачи:
написание функций по работе с файлами, выполнение тестировки программы.

За период выполнения практики были освоены основные понятия и технологии сортировки выбором, реализован метод работы с файлами. Во время выполнения работы Шатравин Д.А. показал себя ответственным, добросовестным учеником, знающим свой предмет, имеющим представление о современном состоянии науки, владеющим современными общенаучными знаниями по информатике и вычислительной технике, программированию и сортировке.

За выполнение работы Шатравин Д.А. заслуживает оценки «_____».

Руководитель практики д.т.н., профессор, Зинкин С.А. « _____ » 2023 г.

Содержание

Введение.....	7
1. Постановка задачи.....	8
1.1 Достоинства алгоритма сортировки выбором.....	8
1.2 Недостатки алгоритма сортировки выбором.....	8
1.3 Типичные сценарии применения данного алгоритма.....	8
2. Выбор решения.....	8
3. Описание программы.....	9
4. Иерархическая блок-схема программы.....	10
5. Тестирование программы.....	11
5.1 Тестирование на разных наборах данных.....	11
5.2 Анализ полученных результатов тестирования (анализ работы алгоритма).....	12
6. Отладка.....	12
7. Совместная разработка.....	13
Заключение.....	15
Список используемой литературы.....	16
Приложение А. Листинг программы.....	17
Приложение В. Результаты тестирования программы.....	23

Введение

Python – это универсальный современный язык программирования высокого уровня, к преимуществам которого относят высокую производительность программных решений и структурированный, хорошо читаемый код. Синтаксис Питона максимально облегчен, что позволяет выучить его за сравнительно короткое время. Ядро имеет очень удобную структуру, а широкий перечень встроенных библиотек позволяет применять внушительный набор полезных функций и возможностей. Он может использоваться для написания прикладных приложений, а также разработки WEB-сервисов.

В конце восьмидесятых голландский программист Гвидо ван Россум работал в научно-исследовательском центре математики и информатики, в котором создавал язык программирования **ABC**. Гвидо вкладывал все силы в этот проект и многому научился в работе над ним. В декабре 1989 года офис закрылся на рождественские каникулы, и Гвидо было нечем заняться. Тогда он просто ради развлечения написал интерпретатор для нового языка программирования. Свой несерьезный проект он назвал «Пайтон» в честь любимого комедийного шоу «Летающий Цирк Монти Пайтона».

1. Постановка задачи

Поставленная задача: Необходимо заполнить массив случайными или введенными числами n -го количества, записать данные в отдельный файл. После этого выполнить сортировку выбором над данными, находящимися в массиве, вывести отсортированные данные, посчитать время выполнения алгоритма.

Использовать сервис GitHub для совместной работы. Создать и выложить коммиты, характеризующие действия, выполненные каждым участником бригады.

Оформить отчет по проведенной практике.

1.1 Достоинства алгоритма сортировки выбором

- Работает хорошо со списками, где элементы расположены случайно или в прямом порядке
- Простая реализация алгоритма
- Быстро сортирует небольшие массивы

1.2 Недостатки алгоритма сортировки выбором

- Высокая алгоритмическая сложность $O(n^2)$
- Медленная сортировка больших массивов
- Медленная сортировка для обратно сортированных списков

1.3 Типичные сценарии применения данного алгоритма

- Ученики в классе (по среднему баллу, количеству пропусков)
- Учет работников для работодателей (по заработной плате, по году рождения, по количеству лет в компании)
- Товары в магазине (по цене, году выпуска, весу)

2. Выбор решения

Для написания данной программы будет использован язык программирования python. Этот язык программирования является распространенным языком программирования. Python – это простой и

универсальный язык программирования, так же известный скоростью разработки на нем. На python удобно писать небольшие проекты. Благодаря выше описанным преимуществам был выбран данный язык программирования.

3. Описание программы

При запуске программы выводится меню из двух пунктов:

- а) Выполнить сортировку
- б) Сортировка с настройками

После выбора пункта а, начнется сортировка выбором элементов в файле data.json, в течении которой элементы перебираются и производится поиск наименьшего элемента, после того как наименьший элемент найден, он вставляется на свое место и начинается поиск наименьшего элемента из оставшихся чисел, так происходит до тех пор, пока набор входных данных не будет исчерпан. Далее этот массив записывается в файл data.json. Пользователю так же выводятся время, количество элементов в массиве и файл с массивом.

Выбрав пункт б, пользователю выводится очередное меню:

- а) Выполнить сортировку
- б) Изменить количество объектов

Можно изменить количество чисел, которое будет генерироваться, по умолчанию это 10000

- с) Изменить файл

Пользователь может изменить файл загрузки чисел, по умолчанию это data.json

- д) Главное меню

Производится выход в главное меню

е) Задать свой массив

Пользователь может задать свой массив

4. Иерархическая блок-схема программы

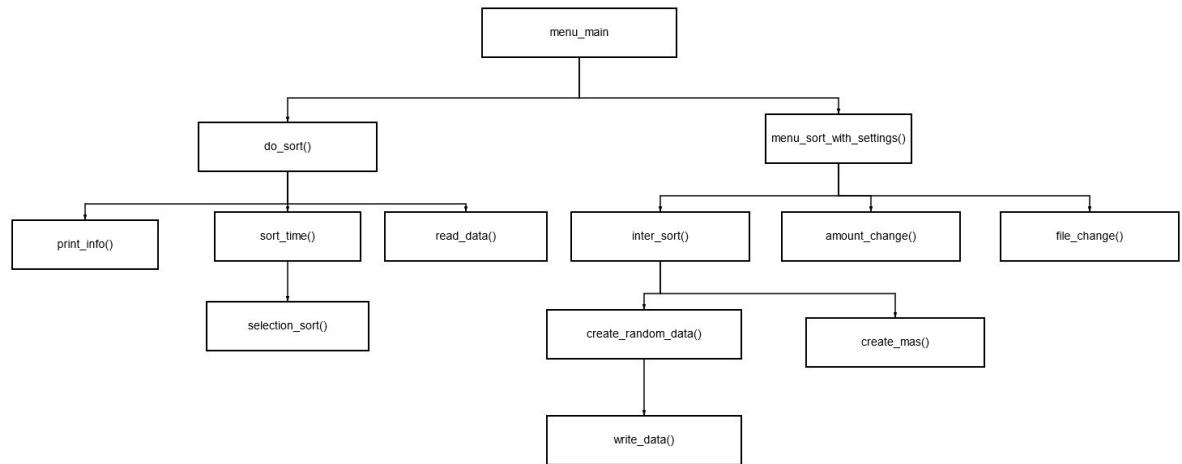


Рисунок 1 – Блок-схема программы

5. Тестирование программы

5.1 Тестирование на разных наборах данных

Тестовый набор данных представлен в таблице 1. Результаты тестирования приведены в Приложении В на рисунках В.1 - В.11.

Таблица 1 – Тестовый набор данных

№ теста	Размер массива size	Время выполнения сортировки в секундах
1	10000	0.15
2	20000	0.62
3	30000	1.40
4	40000	2.49
5	50000	3.92
6	60000	5.60
7	70000	7.63
8	80000	9.98
9	90000	12.63
10	100000	15.58
11	110000	18.86

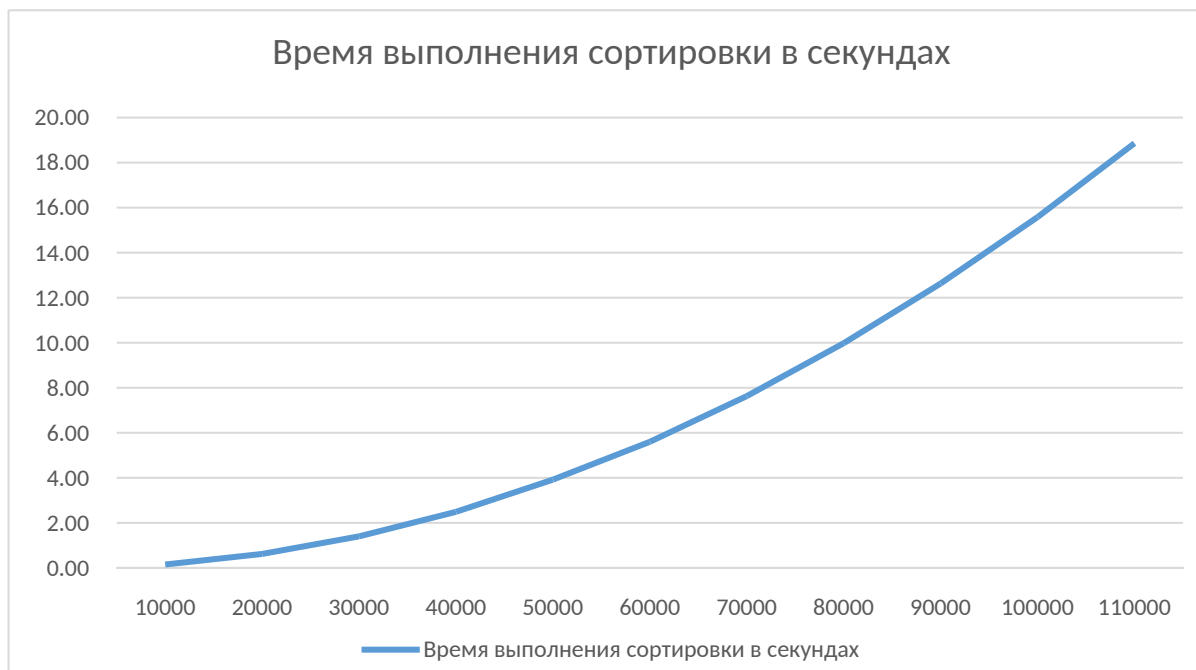


Рисунок 2 – результаты тестирования

5.2 Анализ полученных результатов тестирования (анализ работы алгоритма)

Анализ полученных результатов тестирования (анализ работы алгоритма) На основании анализа данных, полученных в результате тестирования алгоритма сортировки вставками, можно сделать вывод, что время, затраченное на работу программы относительно количества элементов увеличивается линейно, то есть с увеличением количества элементов пропорционально увеличивается время работы программы.

6. Отладка

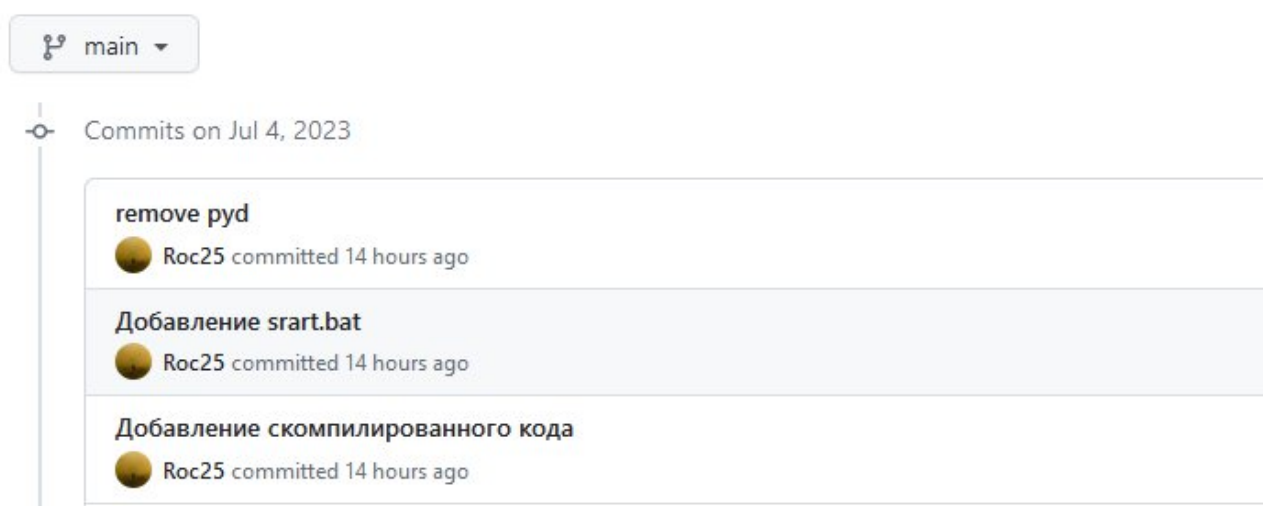
Главная задача отладчика — предоставить возможность заглянуть в процесс выполнения кода. Так, например, можно просмотреть стек вызовов, узнать значения переменных, установить брейкпоинты или запустить выполнение кода построчно.

Для отладки использовалась библиотека `pdb`. Это простая консольная утилита, которая обладает основной функциональностью для отладки кода. Она позволяет как запустить приложение в режиме отладки, без внесения изменений в код, так и поставить свои точки останова.

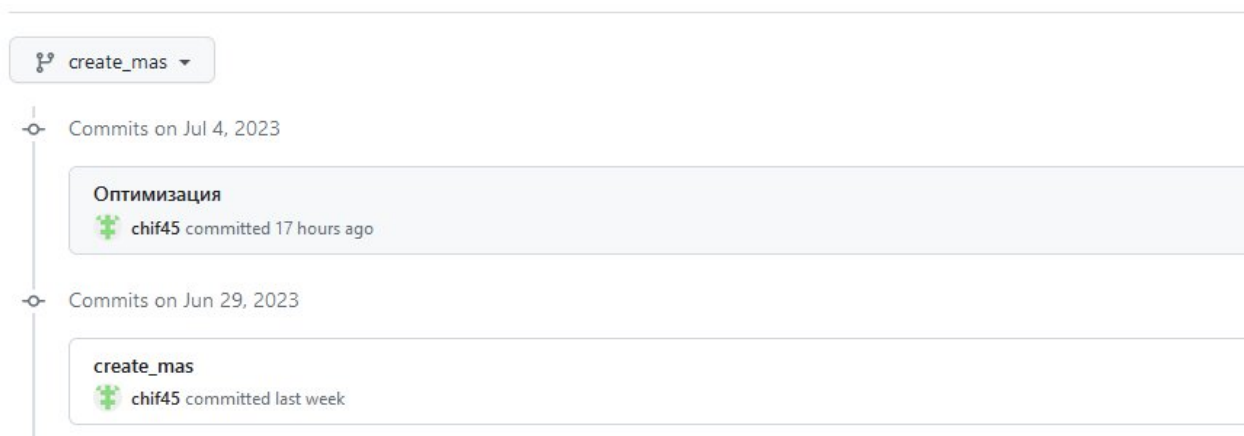
7. Совместная разработка

Во время работы над данной практикой, нашей бригадой осуществлялась совместная работа в GitHub.

Данная программа была написана владельцем репозитория – Поповым Д.А. После написания программы, она была выгружена на удаленный репозиторий Github, на отдельную ветку под названием main.



После этого, второй участник – Перфилов А.В. загрузил данную программу себе на компьютер, с помощью `git clone <ссылка>`, и добавил в нее функцию добавления собственных значений в массив, а также оптимизировал код. После чего создал собственную ветку, и выгрузил обновленный код программы на удаленный репозиторий GitHub.




Далее мной была добавлена функция для работы с файлами, после чего выгружен обновленный код на удаленный репозиторий GitHub.

Shatravin

Commits on Jun 30, 2023

Merge branch 'main' into Shatravin

 babyb0n committed 5 days ago

Оптимизация кода

Dmitriy committed 5 days ago

Оптимизация кода путем добавления cython

Dmitriy committed 5 days ago

Ссылка на удаленный репозиторий:

<https://github.com/Roc25/Practik-2023.git>

Заключение

При выполнении данной работы были получены навыки для совместной работы с помощью GitHub, навыки использования Git Bash. Был изучен алгоритм сортировки выбором.

Мною была написана функция по работе с файлами, а также выполнена тестировка программы.

При выполнении практической работы были улучшены базовые навыки программирования на python и использование сторонних пользовательских библиотек.

В дальнейшем программу можно переписать на ассемблер.

Список используемой литературы

1. <https://habr.com/ru/articles/444594/>
2. <https://docs.cython.org/en/latest/src/tutorial/array.html>
3. <https://habr.com/ru/articles/422085/>

Приложение А. Листинг программы

Файл main.py

```
import time
import json
from pick import pick
from random import randint
from selection_sort import selection_sort

SelectSortBigO = "n^2"

def read_data(file_name: str = "data.json"):
    with open(file_name, "r") as file:
        data = json.load(file)
    return data

def write_data(data: list, file_name: str = "data.json"):
    with open(file_name, "w") as f:
        json.dump(data, f)

def create_mas():
    print("Введите последовательность чисел:")
    list_data = input().split()
    int_lst = []
    for element in list_data:
```

```

    if element.isdigit():
        int_lst.append(int(element))
    else:
        print(f'{element} - не является числом ')
        print('Ошибка формирования списка чисел')
        menu_main()
write_data(int_lst)
menu_main()

```

```

def sort_time(data: list):
    start_time = time.time()
    selection_sort(data, len(data))
    end_time = time.time()

    return end_time-start_time

```

```

def print_info(time_to_sort, file_name, data):
    print(f"Затраченное время: {time_to_sort} сек.")
    print(f"Количество элементов в массиве: {len(data)}")
    print(f"Файл с массивом: {file_name}")

    enter_data = read_data(file_name)
    enter_data.sort()
    print(f"\nПроверка правильности сортировки: {enter_data == data}")

    print(f"Сложность алгоритма по нотации Big O: {SelectSortBigO}")
    input("Для выхода в меню нажмите Enter")

```

```

print("-"*32)
print("")
menu_main()

def do_sort(amount_items: int = 10_000, file_name: str = "data.json"):
    create_random_data(amount_items, file_name)
    data = read_data(file_name)
    time_to_sort = sort_time(data)
    print_info(time_to_sort, file_name, data)

def create_random_data(amount_items: int = 10_000, file_name: str =
"data.json"):
    data_list = []
    for i in range(amount_items):
        data_list.append(randint(-10_000, 10_000))

    write_data(data_list, file_name)

def amount_change(settings: list):
    try:
        inp = int(input("Новое количество элементов в массиве: "))
        settings[0] = inp
    except ValueError:
        print("Ошибка ввода! Введите число")
        amount_change(settings)
    menu_sort_with_settings(settings)

```

```

def file_change(settings: list):
    inp = input("Имя файла: ")
    settings[1] = inp
    menu_sort_with_settings(settings)

def menu_sort_with_settings(settings: list = [10_000, "data.json"]):
    title = "Меню редактирования настроек"

    options = {
        f"Выполнить сортировку": do_sort,
        f"Изменить количество объектов: {settings[0]} (Текущее значение)":
amount_change,
        f"Изменить файл: {settings[1]} (Текущий файл)": file_change,
        f"Главное меню": menu_main,
        f"Задать свой массив": create_mas,
    }

    option, index = pick([key for key, value in options.items()], title)
    if index == 0:
        do_sort(settings[0], settings[1])
    elif index in [1, 2]:
        options[option](settings)
    else:
        options[option]()

```

```

def menu_main():

    title = "При выполнении сортировки создается массив чисел (от -10 000
до 10 000) в файле 'data.json'\n" \

        "Задать количество объектов в массиве и выбрать другой файл
можно с помощью Сортировки с настройками\n"

    options = {
        "Выполнить сортировку": do_sort,
        "Сортировка с настройками": menu_sort_with_settings
    }

    option, index = pick([key for key, value in options.items()], title)
    options[option]()

```

```

if __name__ == '__main__':
    menu_main()

```

Файл Selection_sort.pyx

```

from cpython cimport array
import array

cdef int[:] selection_sort(list a_list, int l):
    cdef int i, j, smallest
    cdef array.array a = array.array('l', a_list)
    cdef int[:] x_list = a
    for i in range(0, l - 1):
        smallest = i
        for j in range(i + 1, l):
            if x_list[j] < x_list[smallest]:

```

```
        smallest = j
    x_list[i], x_list[smallest] = x_list[smallest], x_list[i]
    return x_list
```

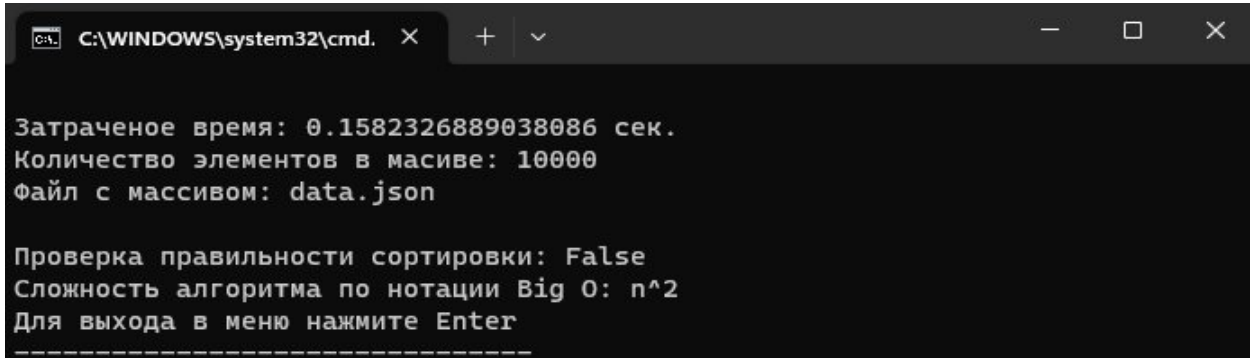
Файл setup.py

```
from distutils.core import setup
from Cython.Build import cythonize
setup(ext_modules=cythonize('selection_sort.pyx'))
```

Файл start.bat

```
python setup.py build_ext --inplace
python main.py
```

Приложение В. Результаты тестирования программы

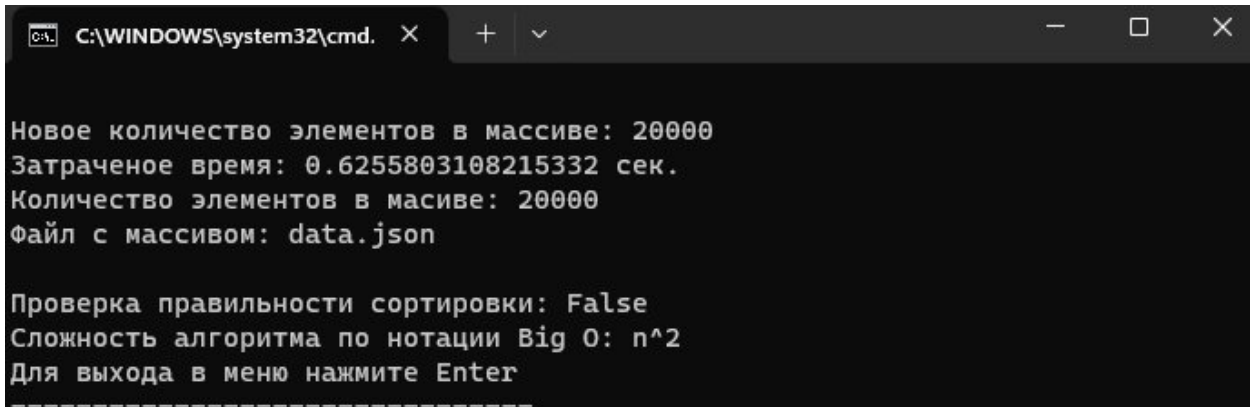


```
C:\WINDOWS\system32\cmd. X + v

Затраченное время: 0.1582326889038086 сек.
Количество элементов в массиве: 10000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O: n^2
Для выхода в меню нажмите Enter
=====
```

Рисунок В.1

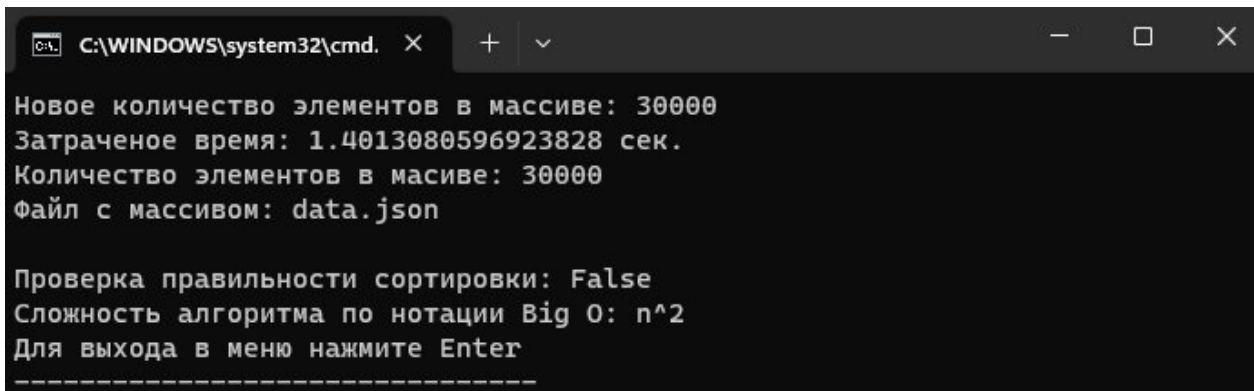


```
C:\WINDOWS\system32\cmd. X + v

Новое количество элементов в массиве: 20000
Затраченное время: 0.6255803108215332 сек.
Количество элементов в массиве: 20000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O: n^2
Для выхода в меню нажмите Enter
=====
```

Рисунок В.2



```
C:\WINDOWS\system32\cmd. X + v

Новое количество элементов в массиве: 30000
Затраченное время: 1.4013080596923828 сек.
Количество элементов в массиве: 30000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O: n^2
Для выхода в меню нажмите Enter
=====
```

Рисунок В.3

```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 40000
Затраченное время: 2.4931342601776123 сек.
Количество элементов в массиве: 40000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.4

```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 50000
Затраченное время: 3.9204301834106445 сек.
Количество элементов в массиве: 50000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.5

```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 60000
Затраченное время: 5.603286981582642 сек.
Количество элементов в массиве: 60000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.6

```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 70000
Затраченное время: 7.630752086639404 сек.
Количество элементов в массиве: 70000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.7


```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 80000
Затраченное время: 9.983458757400513 сек.
Количество элементов в массиве: 80000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.8

```
C:\WINDOWS\system32\cmd. X + v
Новое количество элементов в массиве: 90000
Затраченное время: 12.637635946273804 сек.
Количество элементов в массиве: 90000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.9

```
Новое количество элементов в массиве: 100000
Затраченное время: 15.584949731826782 сек.
Количество элементов в массиве: 100000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter
-----
```

Рисунок В.10

```
Новое количество элементов в массиве: 110000
Затраченное время: 18.861435413360596 сек.
Количество элементов в массиве: 110000
Файл с массивом: data.json

Проверка правильности сортировки: False
Сложность алгоритма по нотации Big O:  $n^2$ 
Для выхода в меню нажмите Enter|
```

Рисунок В.11