

Министерство науки и высшего образования Российской Федерации
федеральное государственное автономное образовательное учреждение
высшего образования «Национальный исследовательский институт
ИТМО»

Факультет МР и П

Алгоритмы и структуры данных

Лабораторная работа №0.

«Введение»

Выполнил студент:

Розметов Джалолиддин

Группа № D3210

Преподаватель: Артамонова Валерия Евгеньевна

г. Санкт-Петербург

2024

Оглавление

Задание 1	2
Реализация задания	3
Задача 1. Код	3
Задача 2. Код	3
Задача 3. Код	4
Задача 4. Код	5
Задание 2	5
Код	6
Задание 3	7
Код	8
Задание 4	9
Время выполнения задание 2. Код	9
Время выполнения Задание 2. Код	11
Вывод	12

Задание 1

Вам необходимо выполнить 4 следующих задачи:

1. Задача $a + b$. В данной задаче требуется вычислить сумму двух заданных чисел. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число — результат сложения $a + b$.
2. Задача $a + b^2$
. В данной задаче требуется вычислить значение $a + b^2$. Вход: одна строка, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$. Выход: единственное целое число — результат сложения $a + b^2$.
3. Выполните задачу $a + b$ с использованием файлов.
 - Имя входного файла: input.txt
 - Имя выходного файла: output.txt

- Формат входного файла. Входной файл состоит из одной строки, которая содержит два целых числа a и b . Для этих чисел выполняются условия $-10^9 \leq a, b \leq 10^9$.
- Формат выходного файла. Выходной файл единственное целое число — результат сложения $a + b$.

Примеры.

input.txt	12 25	130 61
output.txt	37	191

4. Выполните задачу $a+b^2$ с использованием файлов аналогично предыдущему пункту.

Реализация задания

Задача 1. Код

```
while True:
    try:
        values = input("Введите два значения через пробел: ")
        a, b = map(int, values.split())

        if -10**9 <= a <= 10**9 and -10**9 <= b <= 10**9:
            break
        else:
            print("Значение должно быть в диапазоне -10^9 до 10^9");

    except ValueError:
        print("Введено некорректное значение, попробуйте заново")
result = a + b
print(result)
```

Описание кода:

1. Запрашивает у пользователя два целых числа через пробел.
2. Проверяет их находятся ли они в диапазоне от -10^9 до 10^9 .
3. Суммирует введенные числа.
4. Выводит результат.

Задача 2. Код

```
while True:
    try:
        values = input("Введите два значения через пробел: ")
        a, b = map(int, values.split())

        if -10**9 <= a <= 10**9 and -10**9 <= b <= 10**9:
            break
        else:
            print("Значение должно быть в диапазоне -10^9 до 10^9");

    except ValueError:
        print("Введено некорректное значение, попробуйте заново")
```

```
result = a + (b**2)
print(result)
```

Описание кода:

1. Запрашивает у пользователя два целых числа через пробел.
2. Проверяет, находятся ли оба числа в диапазоне от -10^9 до 10^9 .
3. Вычисляет сумму первого числа и квадрата второго числа.
4. Выводит результат.

Задача 3. Код

```
while True:
    try:
        values = input("Введите два значения через пробел: ")
        a, b = map(int, values.split())

        if -10**9 <= a <= 10**9 and -10**9 <= b <= 10**9:
            break
        else:
            print("Значение должно быть в диапазоне -10^9 до 10^9");

    except ValueError:
        print("Введено некорректное значение, попробуйте заново")

with open('input.txt', 'w') as file:
    file.write(values)

result = a + b
with open('output.txt', 'w') as file:
    file.write(str(result))
```

Описание кода:

1. Запрашивает у пользователя два целых числа через пробел.
2. Проверяет, находятся ли оба числа в диапазоне от -10^9 до 10^9 .
3. Записывает введенные значения в файл 'input.txt'.
4. Вычисляет сумму введенных чисел.
5. Записывает результат в файл 'output.txt'.

Описание проведенных тестов:

Тестирование включало проверку правильной обработки корректного ввода двух целых чисел в допустимом диапазоне от -10^9 до 10^9 . Также проводилась проверка обработки некорректного ввода, такого как ввод нецелых чисел или ввод значений вне допустимого диапазона.

Вывод по работе кода:

Код успешно запрашивает у пользователя два целых числа через пробел, проверяет их на корректность и нахождение в заданном диапазоне от -10^9 до 10^9 . Затем он записывает введенные значения в файл 'input.txt' и вычисляет их сумму, записывая результат в файл 'output.txt'.

Задача 4. Код

```
while True:
    try:
        values = input("Введите два значения через пробел: ")
        a, b = map(int, values.split())

        if -10**9 <= a <= 10**9 and -10**9 <= b <= 10**9:
            break
        else:
            print("Значение должно быть в диапазоне -10^9 до 10^9");

    except ValueError:
        print("Введено некорректное значение, попробуйте заново")

with open('input.txt', 'w') as file:
    file.write(values)

result = a + (b**2)
with open('output.txt', 'w') as file:
    file.write(str(result))
```

Описание кода:

1. Запрашивает у пользователя два целых числа через пробел.
2. Проверяет, находятся ли оба числа в диапазоне от -10^9 до 10^9 .
3. Записывает введенные значения в файл 'input.txt'.
4. Вычисляет результат, который является суммой первого числа и квадрата второго числа.
5. Записывает результат в файл 'output.txt'.

Описание проведенных тестов:

Тестирование включало проверку правильной обработки корректного ввода двух целых чисел в допустимом диапазоне, а также обработку некорректного ввода (например, нецелых чисел, пустого ввода и ввода значений за пределами допустимого диапазона).

Вывод по работе кода:

Код успешно реализует запрос двух целых чисел от пользователя, проверяет их корректность и записывает в файл 'input.txt'. Затем он вычисляет сумму первого числа и квадрата второго числа, записывает результат в файл 'output.txt' и завершает работу.

Задание 2

Определение последовательности Фибоначчи

$F_0 = 0$ (1)

$F_1 = 1$

$F_i = F_{i-1} + F_{i-2}$ для $i \geq 2$.

Таким образом, каждое число Фибоначчи представляет собой сумму двух предыдущих, что дает последовательность

0, 1, 1, 2, 3, 5, 8, 13, 21, 34, 55, ...

Ваша цель – разработать эффективный алгоритм для подсчета чисел Фибоначчи. Вам предлагается начальный код на Python, который содержит наивный рекурсивный алгоритм:

```
def calc_fib(n):
    if (n <= 1):
        return n

    return calc_fib(n - 1) + calc_fib(n - 2)

n = int(input())
print(calc_fib(n))
```

- Имя входного файла: input.txt
- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 45$.
- Формат выходного файла. Число F_n .
- Пример.

input.txt	10
output.txt	55

Код.

```
def calc_fib(n):
    fib = [0, 1]
    if 0 <= n <= 45:
        for i in range(2, n + 1):
            fib.append(fib[i - 1] + fib[i - 2])
        return fib[n]
    else:
        print("Значение должно быть в диапазоне 0 до 45")

try:
    with open('input.txt', 'r') as file:
        first_line = file.readline().strip()
        if first_line:
            n = int(first_line)
        else:
            raise ValueError("Файл пустой")
except FileNotFoundError:
    print("Файл не найден")
except ValueError as e:
    print("Ошибка:", e)

try:
    result = calc_fib(n)
except NameError:
    print("Значение не найдено")

try:
```

```
with open('output.txt', 'w') as file:
    file.write(str(result))
except NameError:
    print("Запись в файл не удалась")
```

Вводные данные:

42

Вывод кода:

267914296

Описание кода:

1. Функция `calc_fib(n)` вычисляет n-ное число Фибоначчи для $0 \leq n \leq 45$.
2. Чтение файла: читает число из `input.txt`.
3. Вычисление и запись: вычисляет число Фибоначчи и записывает его в `output.txt`.
4. Обработка ошибок: обрабатывает ошибки при чтении, вычислении и записи.

Описание проведённых тестов:

Тестирование включало проверку правильной работы функции `calc_fib(n)` для значений `n` в диапазоне от 0 до 45, что соответствует корректному расчету чисел Фибоначчи. Также проводилась проверка обработки исключений при чтении файла `'input.txt'`, включая сценарии пустого файла, отсутствия файла и некорректного содержимого файла.

Вывод по работе кода:

Код успешно читает значение `n` из файла `'input.txt'`, вычисляет соответствующее число Фибоначчи с помощью функции `calc_fib(n)` и записывает результат в файл `'output.txt'`. Он также обрабатывает различные ошибочные ситуации, такие как отсутствие файла или некорректные данные в нем.

Задание 3

Определение последней цифры большого числа Фибоначчи. Числа Фибоначчи растут экспоненциально. Например,

$F_{200} = 280571172992510140037611932413038677189525$

Хранить такие суммы в массиве, и при этом подсчитывать сумму, будет достаточно долго. Найти последнюю цифру любого числа достаточно просто: $F \bmod 10$.

- Имя входного файла: `input.txt`

- Имя выходного файла: output.txt
- Формат входного файла. Целое число n . $0 \leq n \leq 10^7$.
- Формат выходного файла. Одна последняя цифра числа F_n .

- Пример 1.

input.txt	331
output.txt	9

$$F_{331} = 668996615388005031531000081241745415306766517246774551964595292186469.$$

- Пример 2.

input.txt	327305
output.txt	5

Это число не влезет в страницу, но оканчивается действительно на 5.

- Ограничение по времени: 5сек.
- Ограничение по памяти: 512 мб.

Код

```
def last_digit_of_fib(n):
    if 0 <= n <= 10**7:
        a, b = 0, 1
        for _ in range(n):
            a, b = b % 10, (a + b) % 10
        return a
    else:
        print()

try:
    with open('input.txt', 'r') as file:
        first_line = file.readline().strip()
        if first_line:
            n = int(first_line)
        else:
            raise ValueError("Файл пустой")
except FileNotFoundError:
    print("Файл не найден")
except ValueError as e:
    print("Ошибка:", e)

try:
    result = last_digit_of_fib(n)
except NameError:
    print("Значение не найдено")

try:
    with open('output.txt', 'w') as file:
        file.write(str(result))
except NameError:
    print("Запись в файл не удалась")
```

Вводные данные:

Вывод кода:

6

Описание кода:

1. Функция `last_digit_of_fib` вычисляет последнюю цифру n-ного числа Фибоначчи для $0 \leq n \leq 10^{**7}$.
2. Чтение файла: читает число из `input.txt`.
3. Вычисление и запись: вычисляет последнюю цифру числа Фибоначчи и записывает её в `output.txt`.
4. Обработка ошибок: обрабатывает ошибки при чтении, вычислении и записи.

Описание проведенных тестов:

Тестирование включало проверку правильного вычисления последней цифры числа Фибоначчи для допустимых значений, обработку ошибок пустого файла, отсутствия файла, некорректных данных и значений вне допустимого диапазона.

Вывод по работе кода:

Код успешно считывает число из файла `input.txt`, вычисляет последнюю цифру n-го числа Фибоначчи, и записывает результат в файл `output.txt`. Он корректно обрабатывает ошибки, связанные с пустым файлом, отсутствием файла, некорректными данными и значениями вне допустимого диапазона.

Задание 4

Задача: вам необходимо протестировать время выполнения вашего алгоритма в Задании 2 и Задании 3.

Дополнительно: вы можете протестировать объем используемой памяти при выполнении вашего алгоритма.

Время выполнения задание 2. Код

```
import time

t_start = time.perf_counter()

def calc_fib(n):
    fib = [0, 1]
    if 0 <= n <= 45:
        for i in range(2, n + 1):
            fib.append(fib[i - 1] + fib[i - 2])
        return fib[n]
    else:
        print("Значение должно быть в диапазоне от 0 до 45")
        return None

try:
    with open('input_ex_2.txt', 'r') as file:
        first_line = file.readline().strip()
        if first_line:
            n = int(first_line)
        else:
```

```

        raise ValueError("Файл пустой")
except FileNotFoundError:
    print("Файл не найден")
    n = None
except ValueError as e:
    print("Ошибка:", e)
    n = None

if n is not None:
    result = calc_fib(n)
else:
    result = None

if result is not None:
    try:
        with open('output_ex_2.txt', 'w') as file:
            file.write(str(result))
    except IOError as e:
        print("Запись в файл не удалась:", e)

print("Время работы: %s секунд" % (time.perf_counter() - t_start))

```

Вывод кода:

C:\Users\71642805\PycharmProjects\Algorithm_Labs\.venv\Scripts\python.exe
C:\Users\71642805\PycharmProjects\Algorithm_Labs\Labs\Semester_1_Labs\exercise_lab0\Lab_0\exercise_4\Testing_ex_2.py

Время работы: 0.002007999981287867 секунд

Process finished with exit code 0

Описание кода:

1. Читает число n из файла input_ex_2.txt.
2. Вычисляет n-е число Фибоначчи, если n в диапазоне от 0 до 45.
3. Записывает результат в файл output_ex_2.txt.
4. Измеряет и выводит время выполнения программы.

Описание проводимых тестов:

Тестирование включало замер времени выполнения кода с использованием функции `time.perf_counter()`, чтобы оценить время выполнения программы в секундах.

Вывод по работе кода:

Код эффективно считывает значение из файла, вычисляет число Фибоначчи и записывает результат, при этом аккуратно управляя ошибками. Функция `time.perf_counter()`, замеряет время выполнения программы в секундах, демонстрируя её производительность в контексте обработки больших данных.

Время выполнения Задание 2. Код

```
import time
t_start = time.perf_counter()
def last_digit_of_fib(n):
    if 0 <= n <= 10**7:
        a, b = 0, 1
        for _ in range(n):
            a, b = b % 10, (a + b) % 10
        return a
    else:
        print()

try:
    with open('input_ex_3.txt', 'r') as file:
        first_line = file.readline().strip()
        if first_line:
            n = int(first_line)
        else:
            raise ValueError("Файл пустой")
except FileNotFoundError:
    print("Файл не найден")
except ValueError as e:
    print("Ошибка:", e)

try:
    result = last_digit_of_fib(n)
except NameError:
    print("Значение не найдено")

try:
    with open('output_ex_3.txt', 'w') as file:
        file.write(str(result))
except NameError:
    print("Запись в файл не удалась")

print("Время рвботы: %s секунд " % (time.perf_counter() - t_start))
```

Вывод кода:

C:\Users\71642805\PycharmProjects\Algorithm_Labs\.venv\Scripts\python.exe

C:\Users\71642805\PycharmProjects\Algorithm_Labs\Labs\Semester_1_Labs\exercise_lab0\Lab_0\exercise_4\Testing_ex_3.py

Время рвботы: 0.001959000015631318 секунд

Process finished with exit code 0

Описание кода:

1. Вычисляет последнюю цифру n-го числа Фибоначчи.
2. Считывает значение n из файла input_ex_3.txt.
3. Записывает результат в файл output_ex_3.txt.
4. Выводит время выполнения в секундах.

Описание проводимых тестов:

Тестирование включало замер времени выполнения кода с использованием функции `time.perf_counter()`, чтобы оценить время выполнения программы в секундах.

Вывод по работе кода:

Код успешно вычисляет последнюю цифру n -го числа Фибоначчи для значения n в диапазоне от 0 до 10^7 , считывая значение из файла 'input_ex_3.txt' и записывая результат в файл 'output_ex_3.txt'. Функция `time.perf_counter()`, замеряет время выполнения программы в секундах, что демонстрирует производительность кода при обработке больших объемов данных.

Вывод

В ходе лабораторной работы были изучены и применены алгоритмы и структуры данных, связанные с числами Фибоначчи. Основной акцент был сделан на оптимизированном алгоритме вычисления последней цифры числа Фибоначчи. Код был разработан для эффективного решения задач, включая обработку ошибок и замер времени выполнения. Результатом стало более глубокое понимание алгоритмов и структур данных, а также их применение на практике.