Задание:

Выполнить анализ работы по нахождению n члена последовательности Фибоначчи с помощью рекурсивного и итерационного подхода.

Решение:

import time

import matplotlib.pyplot as plt

def RecursionMethod(n : int, t: int):

    if n == 0:

        return 0, t + 2

    if n == 1:

        return 1, t + 2

    t += 1

    a, t1 = RecursionMethod(n - 1, t)

    b, t2 = RecursionMethod(n - 2, t)

    return a + b, t1 + t2 + t

def IterationMethod(n : int, t: int):

    t += 1

    first = 1

    second = 1

    t += 1

    for i in range(2, n):

        t += 2

        first, second = first + second, first

    t += 1

    return first, t

def main(n):

    recurrent\_result\_time = []

    recurrent\_result\_iteration = []

    recursion\_result\_time = []

    recursion\_result\_iteration = []

    for i in range(n):

        t = 0

        start\_time = time.time()

        code\_to\_test, iter\_result = IterationMethod(i, t)

        stop\_time = time.time()

        recurrent\_result\_time.append((stop\_time - start\_time) \* 1000)

        recurrent\_result\_iteration.append(iter\_result)

        t = 0

        start\_time = time.time()

        code\_to\_test, iter\_result = RecursionMethod(i, t)

        stop\_time = time.time()

        recursion\_result\_time.append((stop\_time - start\_time) \* 1000)

        recursion\_result\_iteration.append(iter\_result)

    plt.clf()

    plt.title("Временный анализ")

    plt.plot(range(n), recurrent\_result\_time, color="red")

    plt.plot(range(n), recursion\_result\_time, color="black")

    plt.xticks(range(n))

    plt.xlabel("Вычисление номера элемента Фиббоначчи")

    plt.ylabel("Время, мсек")

    plt.legend(["Итерация", "Рекурсия"])

    plt.savefig("Lab1/time.png")

    plt.clf()

    plt.title("Анализ сложности")

    plt.plot(range(n), recurrent\_result\_iteration, color="red")

    plt.plot(range(n), recursion\_result\_iteration, color="black")

    plt.xticks(range(n))

    plt.xlabel("Вычисление номера элемента Фиббоначчи")

    plt.ylabel("Количество операций")

    plt.legend(["Итерация", "Рекурсия"])

    plt.savefig("Lab1/iteration.png")

if \_\_name\_\_ == "\_\_main\_\_":

    main(20)

На рисунке 1 представлен график зависимости времени от нахождения номера последовательности:

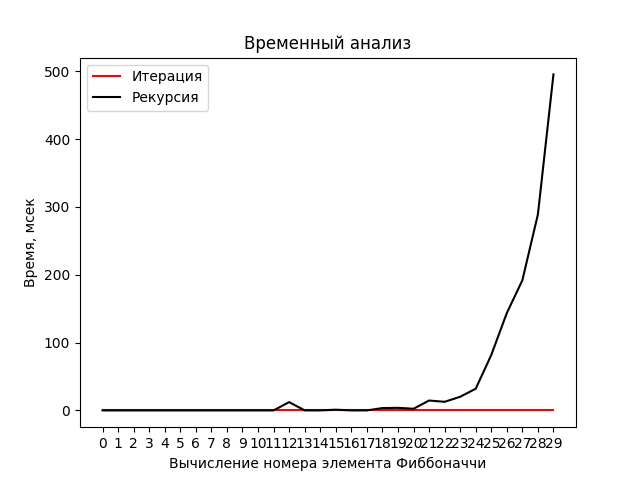


Рисунок 1 – Зависимость времени от номера.

На рисунке 2 представлен график зависимости количества проведённых операций вычисления от номера последовательности:

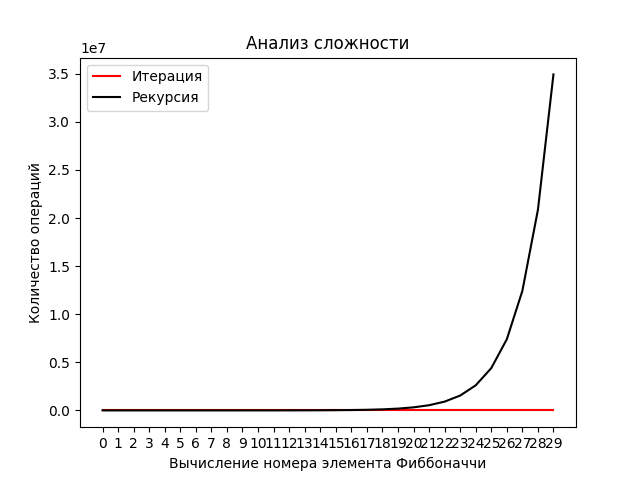


Рисунок 2 – Зависимость операций от номера.