Ульяновский государственный технический университет

Факультет информационных систем и технологий

**Кафедра «Измерительно-вычислительные комплексы»**

Дисциплина «Алгоритмы и структуры данных»

Лабораторная работа №5

Вариант №1

Выполнил: студент группы ИСТбд-13

Абдулкаев Т. Р.

Проверил: преподаватель

Шишкин В. В.

Ульяновск

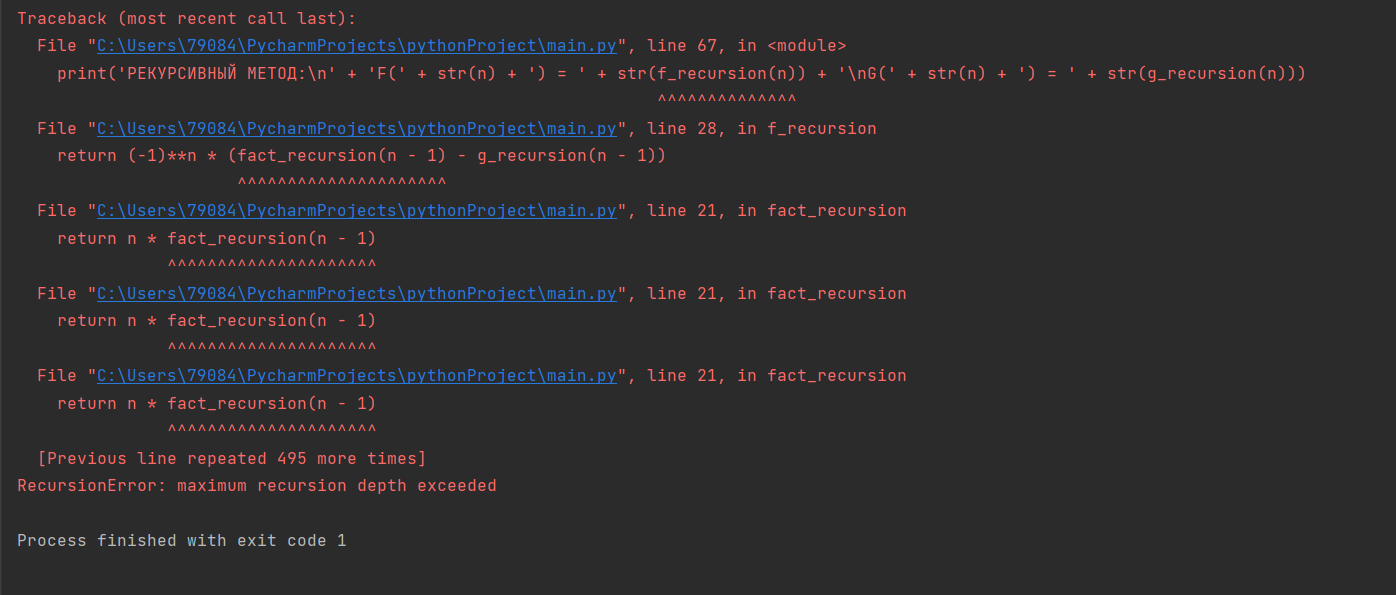
2024

Задание: Задана рекуррентная функция. Область определения функции – натуральные числа. Написать программу сравнительного вычисления данной функции рекурсивно и итерационно. Определить границы применимости рекурсивного и итерационного подхода. Результаты сравнительного исследования времени вычисления представить в табличной и графической форме в виде отчета по лабораторной работе.

Варианты:

1. F(1) = 2; G(1) = 1; F(n) = (-1)n\*( (n–1)! – G(n–1)), G(n) = F(n–1) + G(n–1), при n >=2

Для рекурсивного подхода я написал три функции fact\_recursion (вычисление факториала рекурсивно), f\_recursion (вычисление функции F(n)), g\_recursion (вычисление функции G(n)), каждая из которых кэшируется. Проведя ряд вычислений, меняя значение n, на своем компьютере я дошел до ограничения в самом Python:



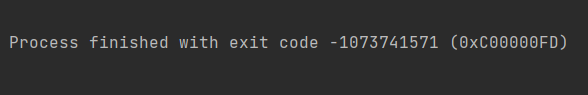
Судя по сообщению, по умолчанию стоит ограничение в 495 повторений цикла, конкретно при вводе значения переменной n с клавиатуры, программа выводит результаты вычислений до n = 499 включительно, далее — ошибка.

Проблему решило:

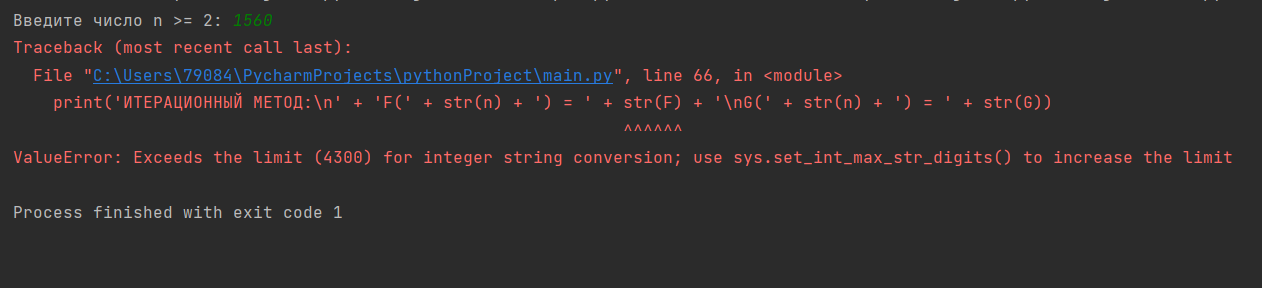
import sys

sys.setrecursionlimit(1000000000)

Что позволило увеличить лимит до n = 1413 включительно, далее — переполнение стека:

Можно было бы решить проблему, увеличив размер стека, но этим я заниматься не стал. Поэтому у меня получилось, что границы рекурсивного метода задаются размером стека и уже далее, возможно, объемом памяти компьютера.

Далее я написал две функции для итерационного метода вычисления: fact\_iteration (вычисление факториала итерационно), f\_g\_iteration (вычисление F(n) и G(n)), каждая из которых также кэшируется. При вводе n = 1560 с клавиатуры, программа вывела сообщение:

Однако с ошибкой написан совет, как решить данную проблему, т. к. ограничение было в конкатенации переменной из int в str.

Добавив в код строку sys.set\_int\_max\_str\_digits(1000000000), проблема исчезла. Далее, чем больше значение n, тем дольше приходится ждать вывода результата. Максимальное значение, которого я дождался на своем компьютере — это n = 10000, программа судя по всему может и дальше, но её выполнение занимает слишком много времени.

Используя функцию timeit() из одноименной библиотеки, я сравнил среднее время выполнения итерационного и рекурсивного метода, где в большинстве случаев был показан результат в пользу рекурсивного метода.

|  |  |  |
| --- | --- | --- |
| № | Итерационно, миллисек. | Рекурсивно, миллисек. |
| 1 | 0.13 | 0.15 |
| 2 | 0.09 | 0.11 |
| 3 | 0.16 | 0.17 |
| 4 | 0.21 | 0.08 |
| 5 | 0.14 | 0.15 |
| 6 | 0.20 | 0.09 |
| 7 | 0.09 | 0.08 |
| 8 | 0.12 | 0.08 |
| 9 | 0.08 | 0.07 |
| 10 | 0.09 | 0.08 |