МИНИСТЕРСТВО ОБРАЗОВАНИЯ И НАУКИ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное

учреждение высшего образования

**«УЛЬЯНОВСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**ОТЧЕТЫ ПО ЛАБОРАТОРНЫМ РАБОТАМ**

**Дисциплина:** Алгоритмы и структуры данных

**Преподаватель:** Шишкин Вадим Викторинович

**Студент:** Клянченков Михаил Андреевич

**Группа:** ИСТбд-11

**г. Ульяновск**

**2024**

**Лабораторная работа №5**

**Вариант 14**

Для данной лабораторной работы был разработан программный код на языке Python, который позволяет сравнить эффективность трех различных подходов к вычислению факториала: рекурсивного, итеративного и динамического. Для этого были реализованы три функции, каждая из которых рассчитывает факториал числа n своим методом, и было измерено время выполнения каждого из этих методов при различных значениях n. Кроме того, результаты измерений были визуализированы с помощью графика для наглядного сравнения.

**Решение:**

В программном коде были определены три функции для вычисления факториала: recursive\_factorial(n), iterative\_factorial(n) и dynamic\_factorial(n). Функция recursive\_factorial(n) рекурсивно умножает число n на результат вызова самой себя с аргументом n-1, пока n не станет равным нулю. Функция iterative\_factorial(n) итеративно умножает числа от 1 до n, последовательно накапливая результат. Функция dynamic\_factorial(n) использует кэш для хранения и повторного использования ранее вычисленных значений факториалов, что сокращает количество вычислений.

Для измерения времени выполнения каждой из функций использовалась функция score\_time(func, n), которая возвращает время, затраченное на 1000 вызовов переданной функции func с аргументом n. Измерения проводились для значений n от 1 до 9.

**Расчеты:**

Используя модуль **timeit** для точного измерения времени выполнения и модуль **matplotlib** для визуализации, были получены следующие результаты:

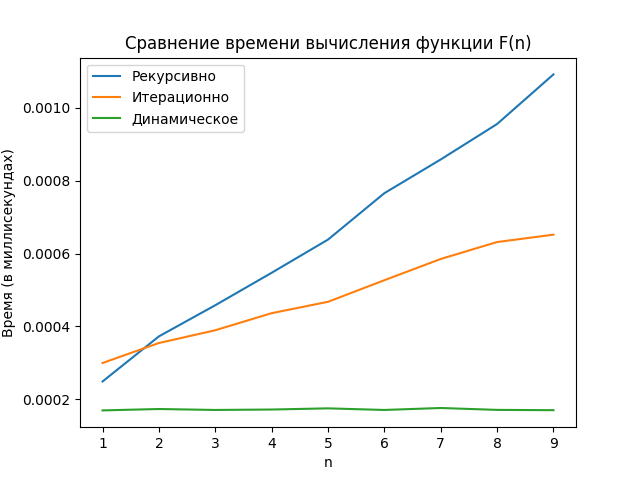
* Время выполнения рекурсивного метода растет экспоненциально с увеличением **n**.
* Итеративный метод показывает линейное увеличение времени выполнения.
* Динамический метод демонстрирует наименьшее время выполнения благодаря мемоизации предыдущих результатов.

**Выводы по лабораторной работе:**

По результатам лабораторной работы было установлено, что динамический подход к вычислению факториала является наиболее эффективным в сравнении с рекурсивным и итеративным методами. Мемоизация (использование кеша) значительно уменьшает количество операций, необходимых для вычисления факториала, особенно при больших значениях n. Рекурсивный метод, хотя и является концептуально простым, оказывается наименее эффективным из-за большого количества повторяющихся вычислений и затрат на вызовы функций. Итеративный метод показывает средние результаты, будучи более предсказуемым и стабильным по сравнению с рекурсивным методом.

Визуализация результатов с помощью графиков позволила наглядно сравнить скорость выполнения каждого из методов и подтвердить преимущества динамического подхода. Эти выводы могут быть использованы при выборе метода подсчета факториала.

**Графический отчет к лабораторной работе №5**



**Табличный отчет к лабораторной работе №5**

D:\Python\venv\Scripts\python.exe D:/Python/main.py

n Рекурсивное время (мс) Итерационное время (мс) Динамическое время (мс)

1 0.0002489000326022506 0.00029959995299577713 0.00016950001008808613

2 0.0003727000439539552 0.00035450002178549767 0.00017330003902316093

3 0.00045819999650120735 0.00038949993904680014 0.00017070001922547817

4 0.0005472999764606357 0.0004363999469205737 0.00017190002836287022

5 0.0006383999716490507 0.0004675999516621232 0.00017509993631392717

6 0.0007658000104129314 0.0005268999375402927 0.00017070001922547817

7 0.0008583999006077647 0.0005852000322192907 0.00017609994392842054

8 0.0009557000594213605 0.000631700037047267 0.00017090002074837685

9 0.001091900048777461 0.0006518999580293894 0.00017010001465678215

Process finished with exit code 0