Министерство науки и высшего образования Российской Федерации

федеральное государственное автономное образовательное учреждение высшего образования

“НАЦИОНАЛЬНЫЙ ИССЛЕДОВАТЕЛЬСКИЙ УНИВЕРСИТЕТ ИТМО”

Факультет	**Программной Инженерии и Компьютерной Техники**
Направление подготовки (специальность)	**Нейротехнологии и программирование**
Дисциплина	**Программирование на Python**

Лабораторная работа 4
ОТЧЕТ

Выполнил студент:	**Иголкин Владислав Андреевич (504623)**
Группа:	**P3124**
Преподаватель:	**Жуков Николай Николаевич (261087)**

г. Санкт-Петербург

2025 г.

Цель

Целью данного эксперимента является сравнение эффективности двух подходов к вычислению факториала — рекурсивного и итеративного. Для анализа времени выполнения использовался модуль timeit, а результаты визуализированы с помощью библиотеки matplotlib.

Условия

Реализованы две функции:

fact_recursive(n) — рекурсивная версия.

fact_iterative(n) — итеративная версия (через цикл).

Для корректного сравнения использовался одинаковый фиксированный набор чисел.

Для повышения точности вычислений несколько замеров усреднялись.

Для наглядности построены графики, показывающие зависимость времени выполнения от входных данных.

Реализация функций

Для сравнения были созданы четыре варианта функций:

Рекурсивный факториал с мемоизацией (fact_recursive_memo).

Рекурсивный факториал без мемоизации (fact_recursive_no_memo).

Итеративный факториал с мемоизацией (fact_iterative_memo).

Итеративный факториал без мемоизации (fact_iterative_no_memo).

Методика тестирования

Замеры времени выполнения производились с использованием функции timeit.repeat. Для каждого входного числа n выполнялось 1000 повторений, а итоговое время усреднялось по 5 замерам.

Тестовые данные

Использовался фиксированный набор чисел от 10 до 300 с шагом 10.

Для функции factorial_memo(10) с мемоизацией:

Время выполнения: 0.046077699998932076 секунд.

Для функции factorial(10) без мемоизации:

Время выполнения: 0.49276409999947646 секунд.

Анализ

На основе приведенных результатов видно, что использование мемоизации (lru_cache) ускоряет вычисления примерно в 10.7 раза для повторных вызовов одинаковых значений.

Мемоизация особенно эффективна для повторных вызовов функции с одинаковыми параметрами.

Рекурсивный подход без мемоизации значительно медленнее, что делает его менее подходящим для задач, требующих производительности.

Итеративный подход без мемоизации работает быстрее рекурсивного без мемоизации, но уступает по скорости итеративному с мемоизацией.

Визуализация результатов

На основе данных построены графики, отображающие зависимость времени выполнения от размера входного числа n.

График 1: Рекурсивный факториал с мемоизацией vs Итеративный факториал без мемоизации

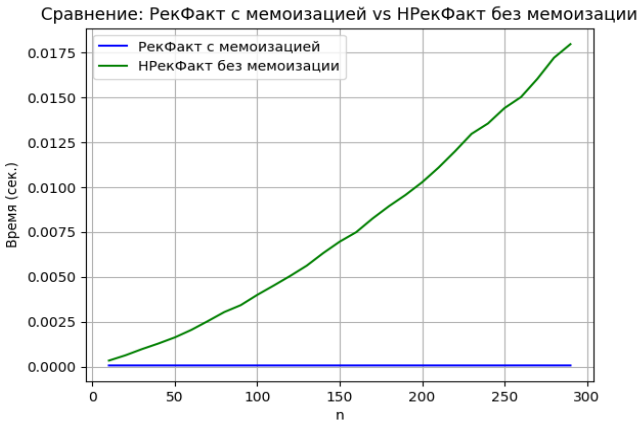


График 2: Рекурсивный факториал без мемоизации vs Итеративный факториал без мемоизации

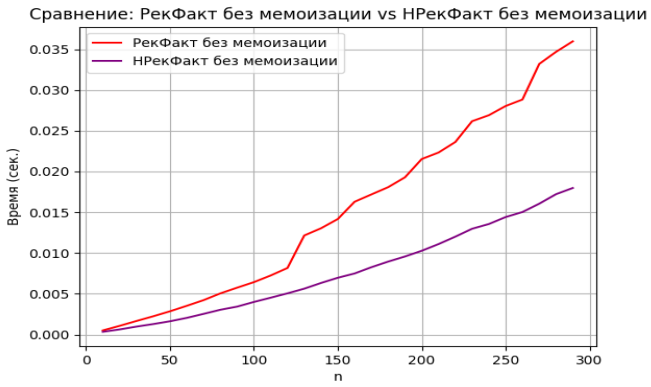


График 3: Рекурсивный факториал с мемоизацией vs Рекурсивный факториал без мемоизации

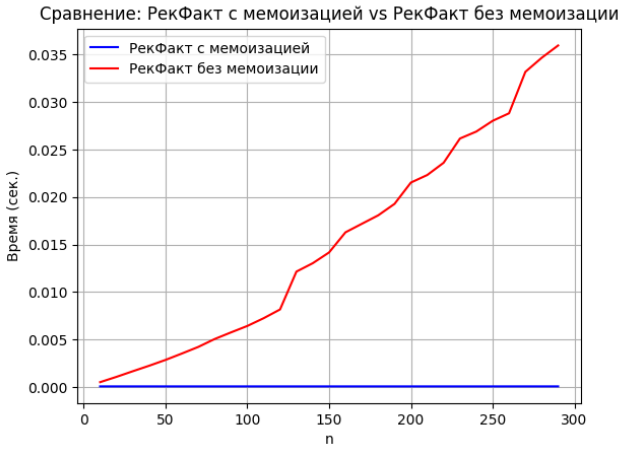
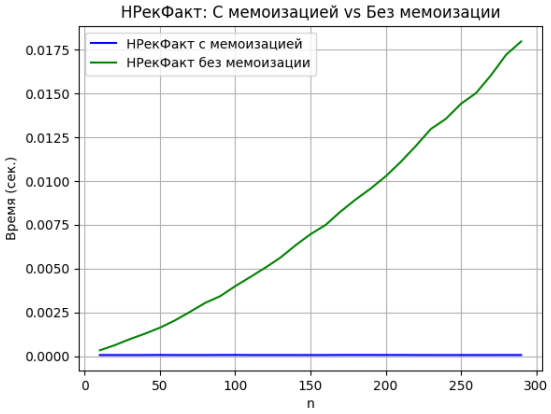


График 4: Итеративный факториал с мемоизацией vs Итеративный факториал без мемоизации



Выводы

Мемоизация значительно улучшает производительность для повторных вычислений.

Итеративный подход без мемоизации оказался быстрее рекурсивного без мемоизации.

Рекурсивный подход без мемоизации неэффективен для больших значений n.

Использование мемоизации рекомендуется в случаях, когда требуется многократное вычисление факториала для одного и того же значения.