

## Тема 04: Хеш-функции и хеш-таблицы

**Цель работы:** Изучить принципы работы хеш-функций и хеш-таблиц. Освоить методы разрешения коллизий. Получить практические навыки реализации хеш-таблицы с различными стратегиями разрешения коллизий. Провести сравнительный анализ эффективности разных методов.

### Теория (кратко):

- **Хеш-функция:** Функция, преобразующая произвольные данные в данные фиксированного размера (хеш-код). Требования: детерминированность, равномерное распределение, скорость вычисления.
- **Хеш-таблица:** Структура данных, реализующая ассоциативный массив. Обеспечивает в среднем  $O(1)$  для операций вставки, поиска и удаления.
- **Коллизия:** Ситуация, когда разные ключи имеют одинаковый хеш-код.
- **Метод цепочек (Chaining):** Каждая ячейка таблицы содержит список элементов с одинаковым хешем. Сложность:  $O(1 + \alpha)$ , где  $\alpha$  - коэффициент заполнения.
- **Открытая адресация (Open Addressing):** Все элементы хранятся в самом массиве. При коллизии ищется следующая свободная ячейка согласно probe sequence.
- **Двойное хеширование (Double Hashing):** Метод открытой адресации, использующий вторую хеш-функцию для определения шага probing.

### Практика (подробно):

#### Задание:

1. Реализовать несколько хеш-функций для строковых ключей.
2. Реализовать хеш-таблицу с методом цепочек.
3. Реализовать хеш-таблицу с открытой адресацией (линейное пробирывание и двойное хеширование).
4. Провести сравнительный анализ эффективности разных методов разрешения коллизий.
5. Исследовать влияние коэффициента заполнения на производительность.

#### Шаги выполнения:

1. **Создание проекта:** Создать файлы `hash_functions.py`, `hash_table_chaining.py`, `hash_table_open_addressing.py`.
2. **Реализация хеш-функций:**
  - Простая хеш-функция (сумма кодов символов)
  - Полиномиальная хеш-функция
  - Хеш-функция DJB2
  - **Для каждой функции указать её особенности и качество распределения.**
3. **Реализация хеш-таблиц:**
  - Метод цепочек с динамическим масштабированием
  - Открытая адресация с линейным пробирыванием
  - Открытая адресация с двойным хешированием
  - **Для каждой операции указать временную сложность в худшем и среднем случае.**
4. **Тестирование:**

- Написать unit-тесты для проверки корректности работы всех реализаций
- Проверить обработку коллизий и переполнения

**5. Экспериментальное исследование:**

- Замерить время выполнения операций для разных коэффициентов заполнения (0.1, 0.5, 0.7, 0.9)
- Исследовать влияние качества хеш-функции на количество коллизий
- **ВАЖНО:** Все замеры проводить на одной вычислительной машине

**6. Визуализация:**

- Построить графики зависимости времени операций от коэффициента заполнения
- Создать гистограммы распределения коллизий для разных хеш-функций

**7. Анализ результатов:**

- Сравнить эффективность разных методов разрешения коллизий
- Определить оптимальный коэффициент заполнения для каждой реализации
- Проанализировать влияние хеш-функции на производительность

**8. Оформление отчета:** Результаты оформить в файле **README.md**. Код должен соответствовать PEP8.

**9. Контроль версий:** Стратегия ветвления - GitHub Flow.

**Критерии оценки:**

• **Оценка «3» (удовлетворительно):**

- Реализована 1 хеш-функция и 1 вариант хеш-таблицы
- В коде присутствуют комментарии с оценкой сложности операций
- Проведены базовые замеры производительности

• **Оценка «4» (хорошо):**

- Выполнены все критерии на «3»
- Реализованы 2 хеш-функции и 2 варианта хеш-таблиц
- Код хорошо оформлен и полностью прокомментирован
- Проведены замеры для разных коэффициентов заполнения
- Построены графики зависимости времени от коэффициента заполнения

• **Оценка «5» (отлично):**

- Выполнены все критерии на «4»
- Приведены характеристики ПК для тестирования
- Реализованы 3+ хеш-функции и 3 варианта хеш-таблиц
- Проведен полный сравнительный анализ всех методов
- Построены гистограммы распределения коллизий
- В отчете присутствует детальный анализ с выводами о применении каждого метода

**Рекомендованная литература**

1. Юрий Петров: "Программирование на Python" — онлайн-курс и учебные материалы.

- Ссылка для изучения: <https://www.yuripetrov.ru/edu/python/index.html>

2. **Кормен, Т., Лейзерсон, Ч., Ривест, Р., Штайн, К.** Алгоритмы: построение и анализ, 3-е издание.  
— М.: Вильямс, 2022. — 1328 с.
  - (*Оригинальное название: Cormen, T.H., Leiserson, C.E., Rivest, R.L., Stein, C. Introduction to Algorithms, 3rd Edition*)
3. **Скиена, С.** Алгоритмы. Руководство по разработке, 3-е издание. — СПб.: БХВ-Петербург, 2022. — 720 с.
  - (*Оригинальное название: Skiena, Steven S. The Algorithm Design Manual, 3rd ed.*)