

Лабораторная работа 4

Хеширование

(20 баллов)

Выполните самостоятельно следующие задания и оформите отчет.

Требования по отчету:

Наличие титульного листа. Размер страницы должен соответствовать формату А4 (210x297), размеры полей: левое – 30 мм, правое – 10 мм, верхнее – 15 мм, нижнее – 20 мм. Шрифт Times new Roman, размер 14 pt полуторный междустрочный интервал. Выравнивание текста – по ширине, красная строка – 1,25 см, отступ слева и справа – 0 мм.

№ 1

(5 балла)

Возьмите реализацию класса `HashTable` из лекционных материалов и выполните следующие доработки:

1. Реализуйте квадратичное пробирование как технику повторного хеширования.
2. Реализуйте работу с функцией `len` (переопределите метод `__len__`).
3. Реализуйте работу оператора `in` (переопределите метод `__contains__`).
4. Переделайте метод `put` таким образом, чтобы таблица автоматически меняла размер, когда загрузочный фактор становится больше значения 0.7. Размер должен увеличиваться примерно в два раза до ближайшего подходящего простого числа.
5. Реализуйте работу оператора `del` (переопределите метод `__delitem__`) для удаления элемента таблицы. Таблица должна автоматически менять размер, когда загрузочный фактор становится меньше значения 0.2. Размер должен уменьшаться примерно в два раза до ближайшего подходящего простого числа.

Все выполненные доработки должны быть протестированы.

№ 2

(5 балла)

Возьмите реализацию класса `HashTable` из лекционных материалов и выполните следующие доработки:

1. Переделайте существующие методы так, чтобы разрешение коллизий происходило не при помощи концепции открытой адресации, а методом цепочек. Для этого в каждом слоте храните связный список, реализованный классом `UnorderedList` из лабораторной работы 3.
2. Реализуйте работу с функцией `len` (переопределите метод `__len__`).
3. Реализуйте работу оператора `in` (переопределите метод `__contains__`).
4. Переделайте метод `put` таким образом, чтобы таблица автоматически меняла размер, когда загрузочный фактор становится больше значения 0.7. Размер должен увеличиваться примерно в два раза до ближайшего подходящего простого числа.
5. Реализуйте работу оператора `del` (переопределите метод `__delitem__`) для удаления элемента таблицы. Таблица должна автоматически менять размер, когда

загрузочный фактор становится меньше значения 0.2. Размер должен уменьшаться примерно в два раза до ближайшего подходящего простого числа.

Все выполненные доработки должны быть протестированы.

№ 3

(2 балла)

Переделайте класс HashTable, чтобы в качестве ключей можно было использовать строки.

№ 4

(2 балла)

Дана строка русского текста, состоящая из слов и пробелов. Словом считается последовательность русских букв, слова разделены одним или большим числом пробелов.

Для каждого слова этого текста узнайте порядковый номер его вхождения в текст именно в той форме, в которой указано слово. Для первого вхождения слова выведите «1», для второго вхождения того же слова выведите «2» и так далее.

Для решения этой задачи используйте класс HashTable из задания № 3.

Пример

Ввод

Раз раз раз как меня слышно Повторяю раз раз раз Повторяю

Вывод

1 1 2 1 1 1 1 3 4 5 2

№ 5

(3 балла)

Напишите программу, имитирующую процесс регистрации и авторизации. Для каждого пользователя программа должна сохранять логин, хеш его пароля и «соль». Для хранения данных можно использовать БД или файл.

Действия при сохранении пароля:

1. Сгенерируйте длинную случайную «соль» при помощи модуля secrets ([secrets — Generate secure random numbers for managing secrets — Python 3.10.0 documentation](#)). Длина «соли» должна быть такой же как и выходные данные используемой вами хэш-функции. Например, если для хеширования вы используете SHA256, то на выходе вы получите 256 бит (32 байта). В этом случае соль должна составлять не менее 32 случайных байт.
2. Добавьте «соль» к паролю и хэшируйте его с помощью функции scrypt из модуля hashlib ([Хеширование паролей модулем hashlib в Python. \(docs-python.ru\)](#)).
3. Сохраните логин, «соль» и получившейся хэш в БД или в файл.

Действия при проверке пароля:

1. Извлеките «соль» и хэш пользователя из БД или файла.
2. Добавьте «соль» к введенному паролю и хэшируйте его, используя ту же хэш-функцию, что и в алгоритме сохранения пароля.

3. Сравните получившейся хэш введенного пароля с хэшем из БД или файла. Если они совпадают, то пароль правильный. В противном случае пароль был введен неверно.

№ 6

(3 балла)

Напишите программу, которая принимает от пользователя путь до директории. Для всех файлов из данной директории должен быть вычислен хэш. Программа должна выявить и вывести на экран все дубликаты в этой директории (т.е. файлы, у которых одинаковый хэш).