Санкт-Петербургский Национальный Исследовательский Университет Информационных Технологий, Механики и Оптики

Факультет Инфокоммуникационных Технологий

**Лабораторная работа №1**

Выполнили:

Новиков Н. В.

Зенин Д.Д.

Проверил Мусаев А. А.

Санкт-Петербург,

2024

**Содержание**

[ВВЕДЕНИЕ 3](#_Toc161852191)

[1. Написание алгоритма с использованием CRC-8 и т.д. 4](#_Toc161852192)

[2. Алгоритм с использованием деления и умножения 4](#_Toc161852193)

[Задача с Leetcode 4](#_Toc161852194)

[ЗАКЛЮЧЕНИЕ 6](#_Toc161852195)

# ВВЕДЕНИЕ

Цель лабораторной работы №1 состояла в знакомстве алгоритмами хэширования. Для достижения этой цели были поставлены следующие задачи:

1. Написать алгоритм с использованием CRC-8, CRC-16, CRC-32

2. Написать алгоритм с использованием умножения и деления

# 1. Написание алгоритма с использованием CRC-8, CRC-16, CRC\_32

# 

Рисунок 1 - Алгоритм с использованием CRC-8, CRC-16, CRC-32

Алгоритм работает следующим образом: введенный текст преобразуется в двоичный код путем преобразования ASCII-кода каждого символа в двоичный код подряд. К полученному значению добавляется количеству нулей, равному степени полинома алгоритма. После полученное значение делится посимвольно, используя XOR к каждому символу, столбиком на двоичный код шестнадцатеричного табличного значения. Результат множественного хэширования и является результатом работы алгоритма.

**Задача с LeetCode**

Задача на использование функции хэширования.

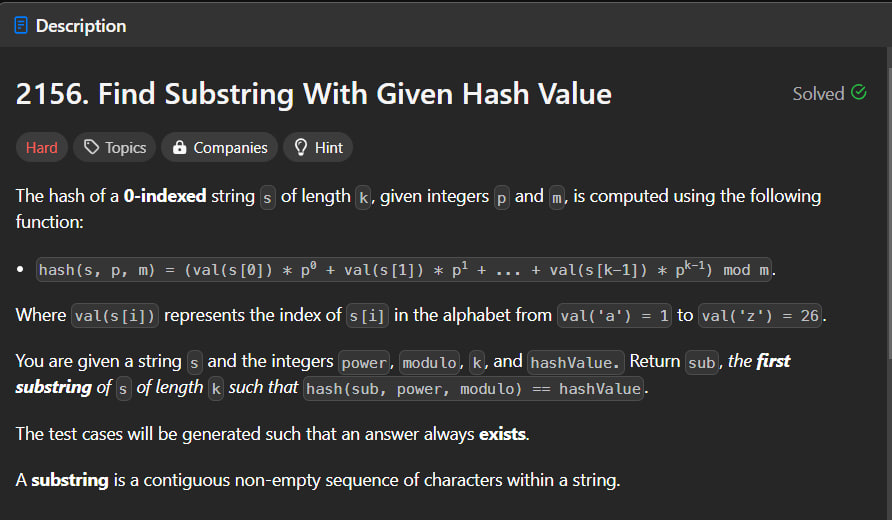


Рисунок 2 – Условие задачи

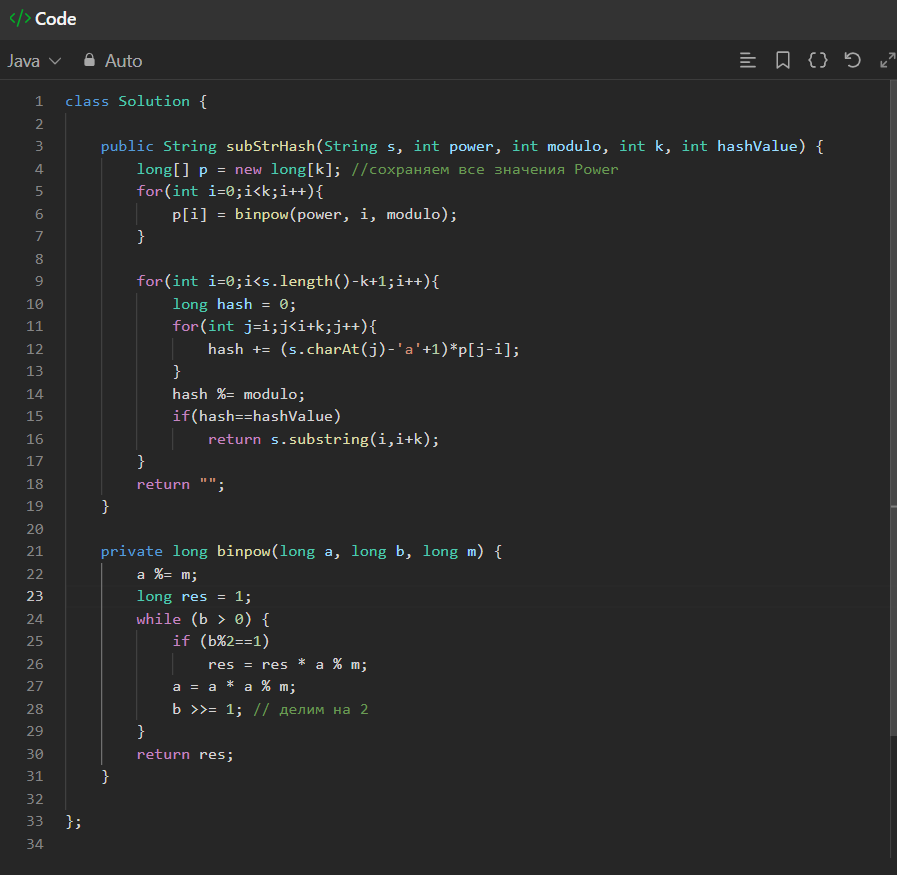


Рисунок 3 – Решение задачи

Задача решена с использованием эффективного двоичного алгоритма возведения в степень. Изначально мы сохраняем все значения возведения числа power в степени, чтобы каждый раз их не считать. После отрезок длиной k двигаем по строке и сравниваем значение из хеш-функции и данное значение hashValue. При совпадении, возвращаем подстроку.

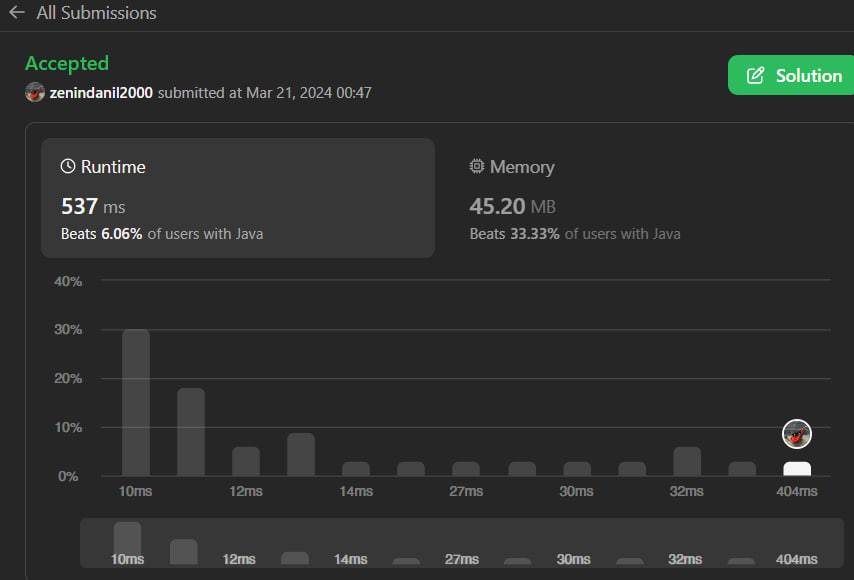


Рисунок 4 – Результаты выполнения задачи

# 2. Алгоритм с использованием деления и умножения

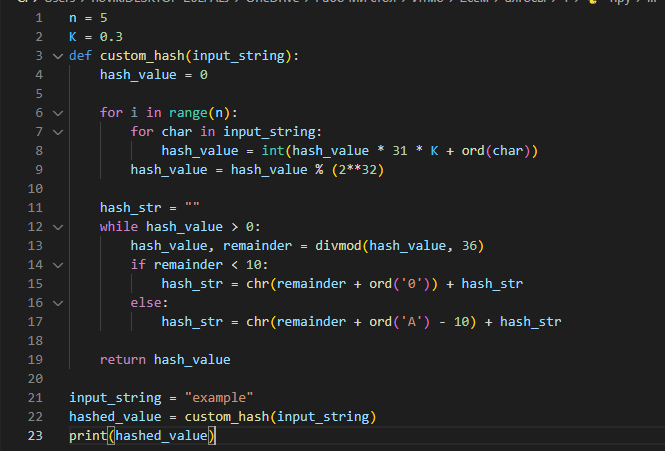


Рисунок 5 - Алгоритм умножения и деления

Алгоритм работает следующим образом. Цикл проходится по каждому символу строки. После находится код ASCII и по методу умножения, а после деления. После для красоты хэширования после преобразования данные преобразуются в буквы.

# Задача с Leetcode

Задача с Leetcode была выбрана из вкладки Hash Table. Решение построено на рекурсивном переборе всей доски. Функция is\_valid проверяет можно ли поставить туда цифру, если да, то ставить и идет дальше. При обнаружении конфликта возвращается назад и начинает по новой.

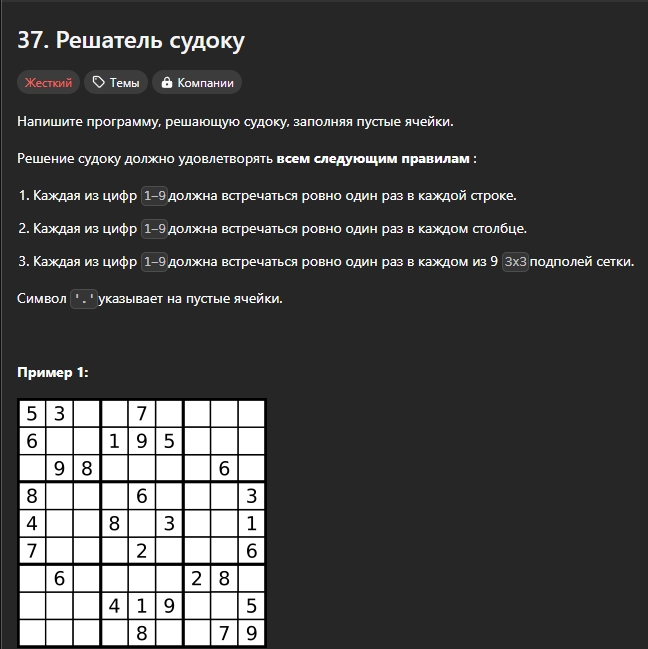


Рисунок 6 - Задача с LeetCode

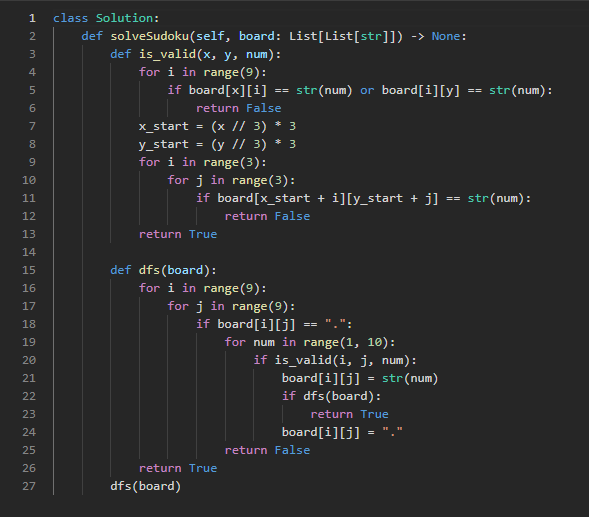


Рисунок 7 - Решение задачи

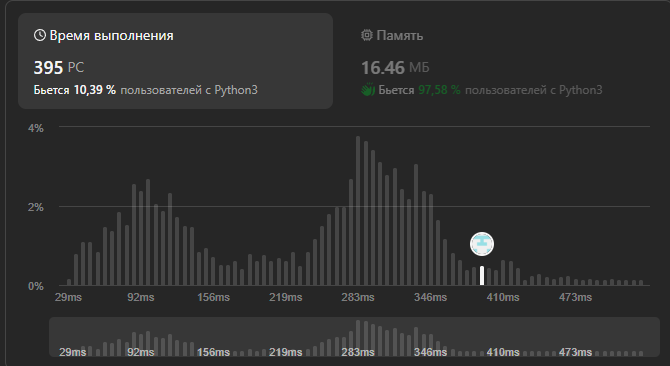


Рисунок 8 - Результаты выполнения

# ЗАКЛЮЧЕНИЕ

В ходе выполнения лабораторной работы №1 были получены навыки работы с алгоритмами хэширования