Правительство Российской Федерации

Федеральное государственное автономное образовательное учреждение высшего профессионального образования «Национальный исследовательский университет «Высшая школа экономики»»

Московский институт электроники и математики Национального исследовательского университета «Высшая школа экономики»

ОТЧЁТ

По лабораторной работе №3

По дисциплине «Методы программирования»

Выполнил:

Ваганов В.Е.

Проверил:

Драчев Г.А.

Москва 2023

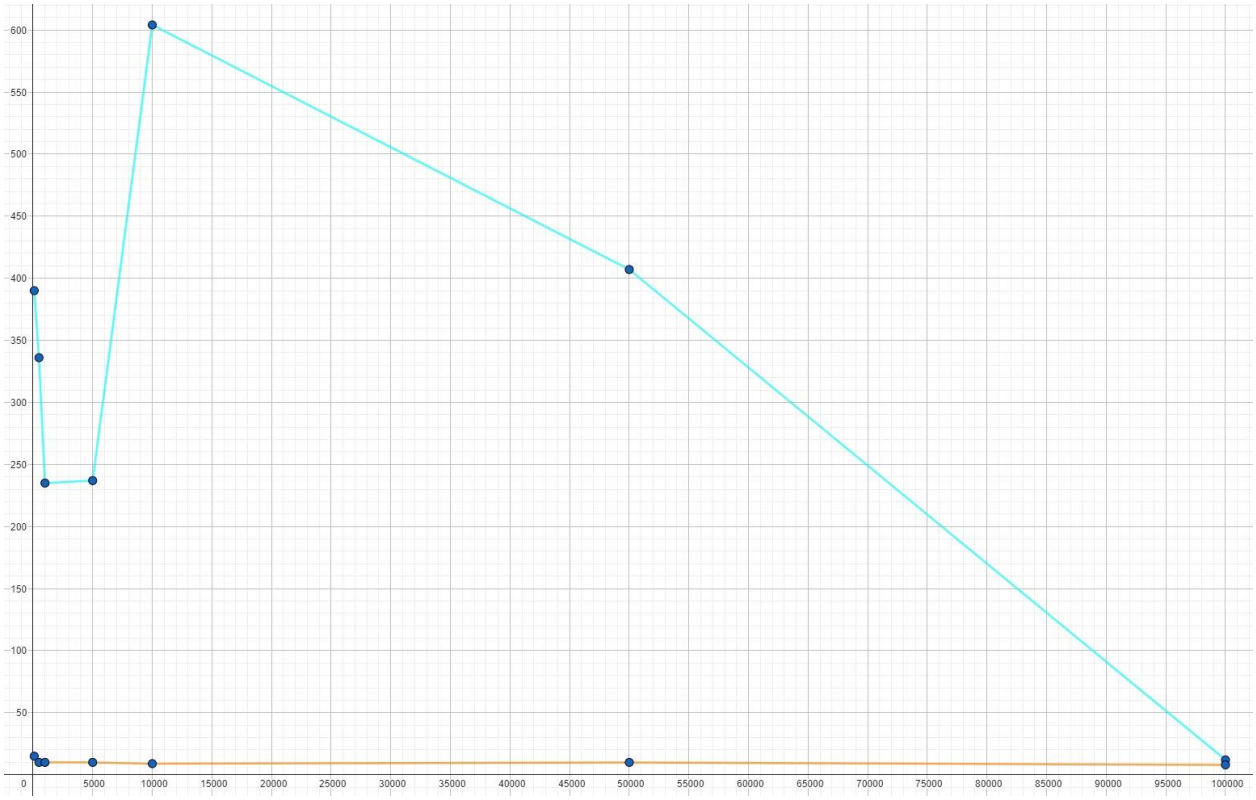
Исходный код лабораторной работы находится по ссылке:

https://github.com/vladimir-vaganov/ProgTech_LAB3/blob/main/Lab_MP3.cpp

Документация в виде html-страницы находится по ссылке:

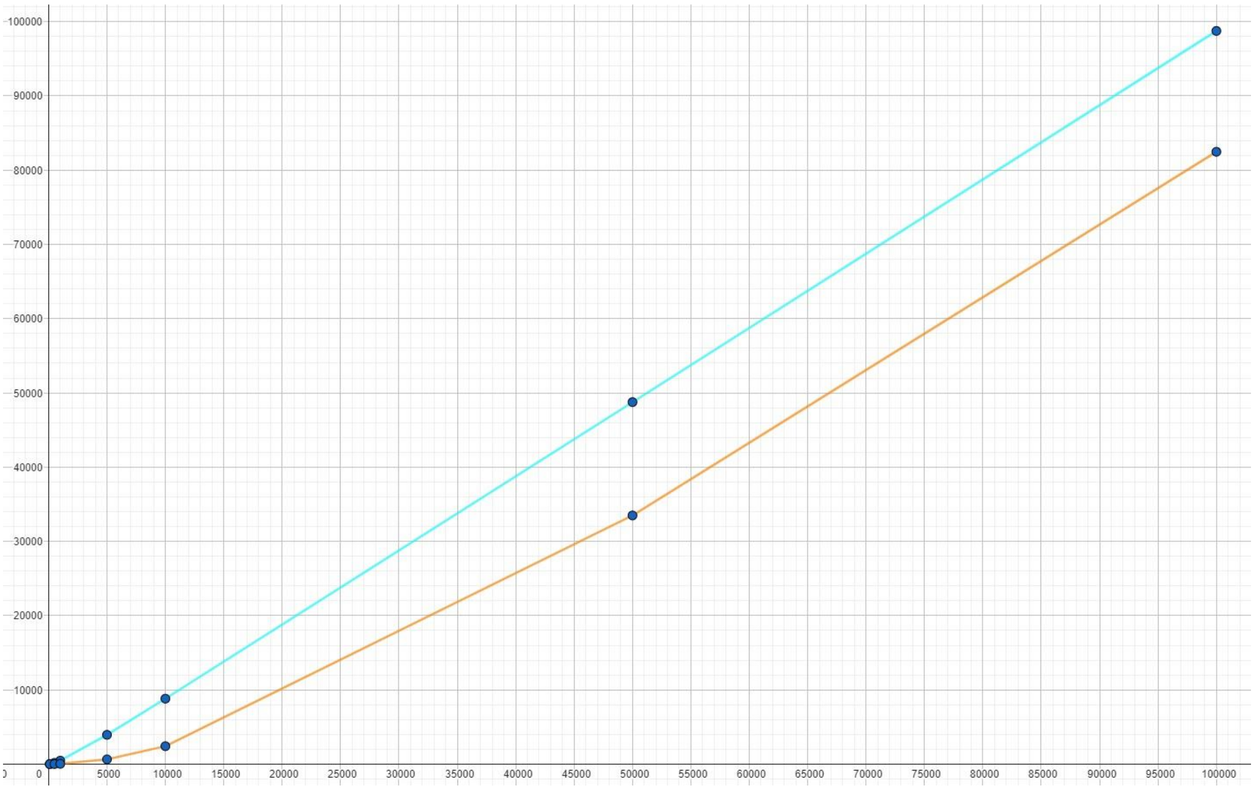
https://github.com/vladimir-vaganov/ProgTech_LAB3/tree/main/html

Графики зависимости работы времени поиска от размерности массива данных (голубой – первая хэш-функция, оранжевый - вторая):



Явно видно, что поиск в хэш-таблице с «плохой» функцией намного дольше и нестабилен. Поиск в хэш-таблице, построенной на функции с меньшим числом коллизий стабилен и подходит под асимптотику O (1).

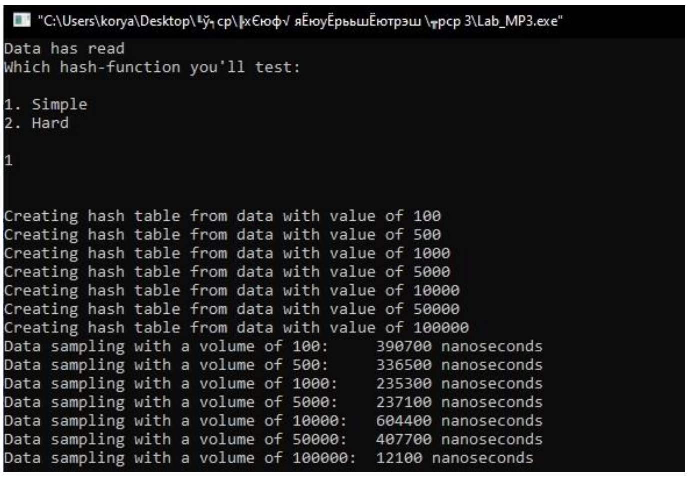
Графики зависимостей числа коллизий от размерности массива:



По графику видно, что у «хорошей» хэш-функции коллизий меньше. Рост графика с ростом размерности массива обосновывается тем, что все данные в массивах состоят из строк длины 3, состоящих из строчных букв латинского алфавита. То есть, максимальное число возможных различных строк – 17576. Таким образом, коллизии будут из-за повторяющихся равных значений.

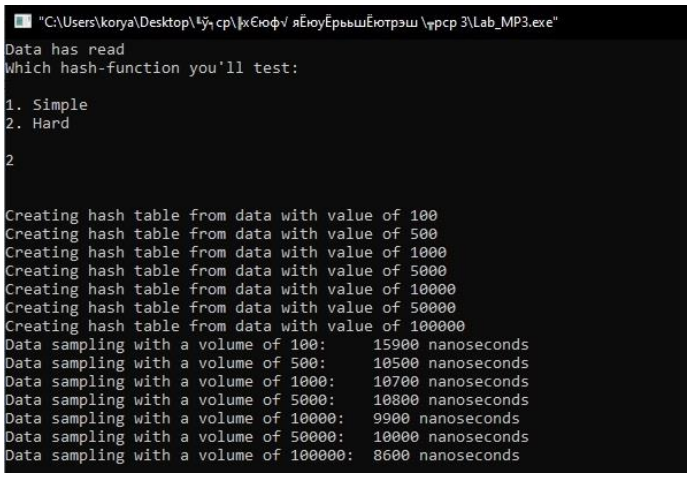
Скриншоты запуска программы:

Время поиска по первой хэш-таблице:

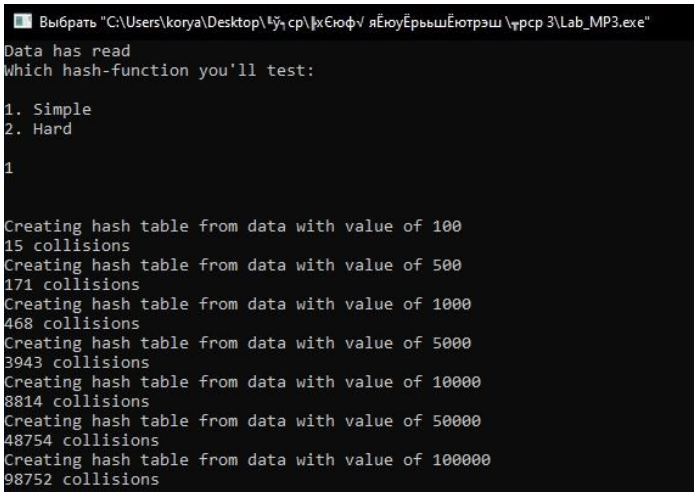


```
"C:\Users\korya\Desktop\ҰўҷсрѤ|х€юф√ лЁюуЁрьшЁютрэш \╓рср 3\Lab_MP3.exe"
Data has read
Which hash-function you'll test:

1. Simple
2. Hard

1


Creating hash table from data with value of 100
Creating hash table from data with value of 500
Creating hash table from data with value of 1000
Creating hash table from data with value of 5000
Creating hash table from data with value of 10000
Creating hash table from data with value of 50000
Creating hash table from data with value of 100000
Data sampling with a volume of 100:     390700 nanoseconds
Data sampling with a volume of 500:     336500 nanoseconds
Data sampling with a volume of 1000:    235300 nanoseconds
Data sampling with a volume of 5000:    237100 nanoseconds
Data sampling with a volume of 10000:   604400 nanoseconds
Data sampling with a volume of 50000:   407700 nanoseconds
Data sampling with a volume of 100000:  12100 nanoseconds
```

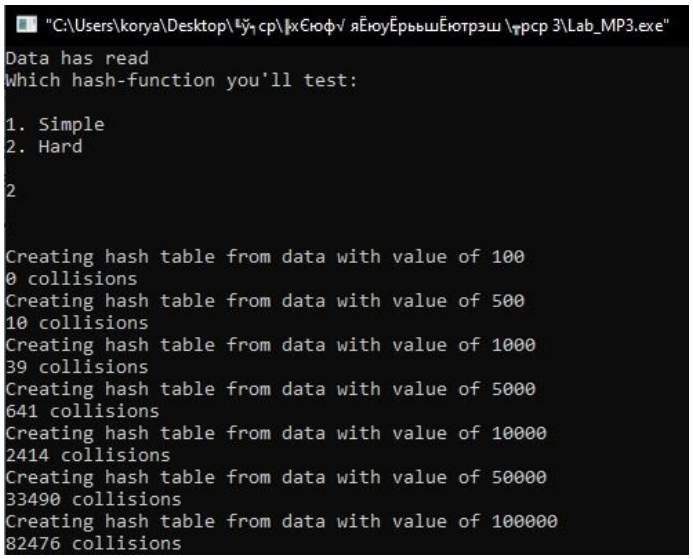
Время поиска по второй хэш-таблице:



Количество коллизий при использовании первой функции:



Количество коллизий при использовании второй функции:



Сравнение с результатами прошлой работы:

При использовании первой хэш-функции время поиска больше, чем при использовании multimap, который является по своей сути такой же хэш-таблицей.

Однако при использовании второй функции асимптотика схожа и время примерно одинаковое, однако всё равно больше. Это можно обосновать внутренними оптимизациями контейнера multimap, а также более оптимальной хэш-функцией.