**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ**

**ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра Вычислительной техники**

**ОТЧЕТ**

**по лабораторной работе № 3**

**по дисциплине «Алгоритмы и структуры данных»**

**Тема: Комбинированные структуры данных и стандартная библиотека шаблонов**

|  |  |  |
| --- | --- | --- |
| Студенты гр. 1305 |  | Данилов Артем и Серкин Данил |
| Преподаватель |  | Колинько П. Г. |

Санкт-Петербург

2023

**Цель работы.**

Изучить различные структуры данных для хранения множеств и последовательностей, в том числе с использованием стандартных библиотек и шаблонов языка С++, научиться работать с такими структурами.

**Задание (вариант 2)**

Реализовать индивидуальное задание темы «Множества + последовательности» в виде программы, используя свой контейнер для заданной структуры данных (хеш-таблицы или одного из вариантов ДДП), и доработать его для поддержки операций с последовательностями. Для операций с контейнером рекомендуется использовать возможности библиотеки алгоритмов. Программа должна реализовывать цепочку операций над множествами. Результат каждого шага цепочки операций выводится на экран.

* Базовая СД: АВЛд
* Дополнительные операции: CONCAT, EXCL, MUL
* Цепочка операций: (A \ B \ C) ⊕ D ∪ E
* Мощность каждого множества: 26

**Класс Tree**

Класс Tree реализует АВЛ-дерево и представляет собой пользовательский контейнер, поддерживающий операции над множествами и последовательностями с использованием библиотеки алгоритмов (например, функции set\_difference, set\_symmetric\_difference, set\_union, remove\_if, find, swap). Каждый узел имеет динамический массив целых чисел (последовательный контейнер vector), представляющий собой массив порядковых номеров вставки ключа в дерево, что позволяет восстанавливать последовательность за время O(n).

Предполагаемое время выполнения операций над множествами - O(log n). Каждая операция над последовательностью включает в себя восстановление последовательности за O(n) и обратное преобразование в дерево за O(n \* log n).

Предполагаемое время выполнения операции CONCAT – O(n \* log n); подсоединение последовательности происходит за O(n).

Предполагаемое время выполнения операции EXCL – O(n^2); исключение второй последовательности из первой происходит за O(n^2).

Предполагаемое время выполнения операции MUL – O(n \* log n); сцепление последовательности самой за собой происходит за O(n).

**Пример работы программы на случайных данных**

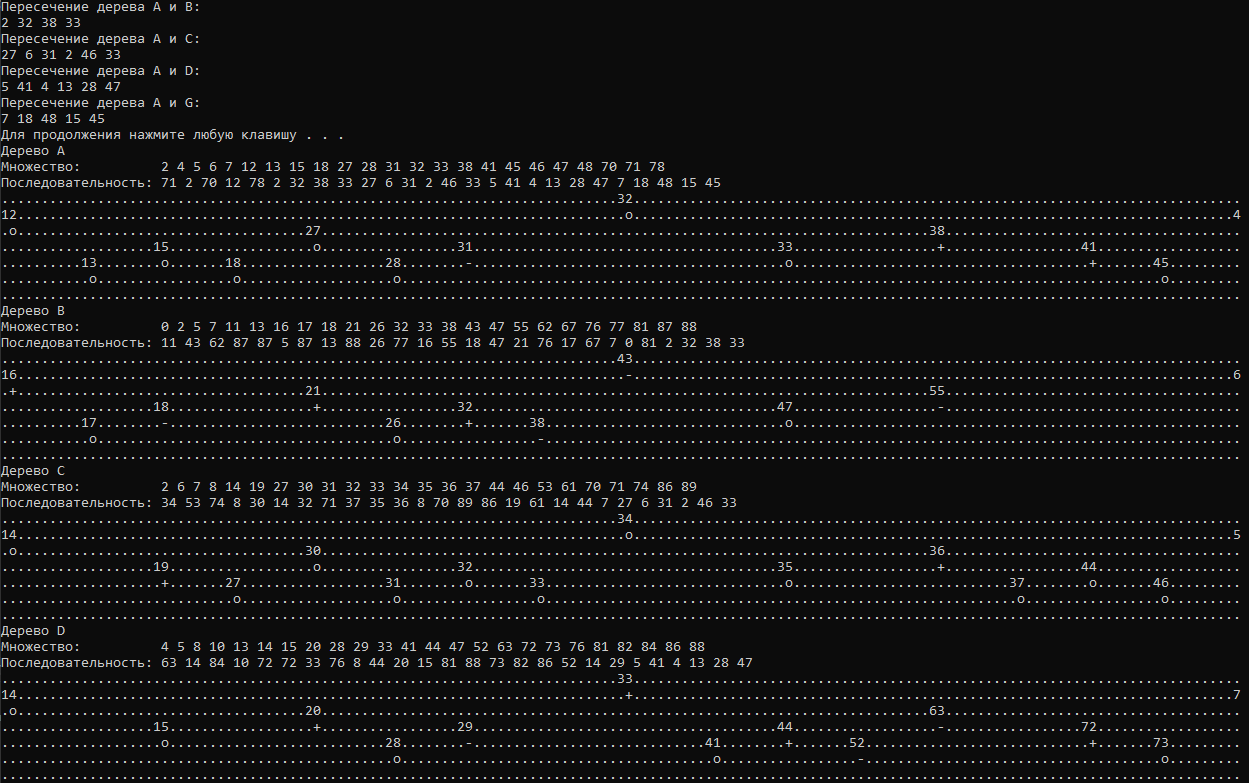


Рис. 1. Деревья A, B, C, D.

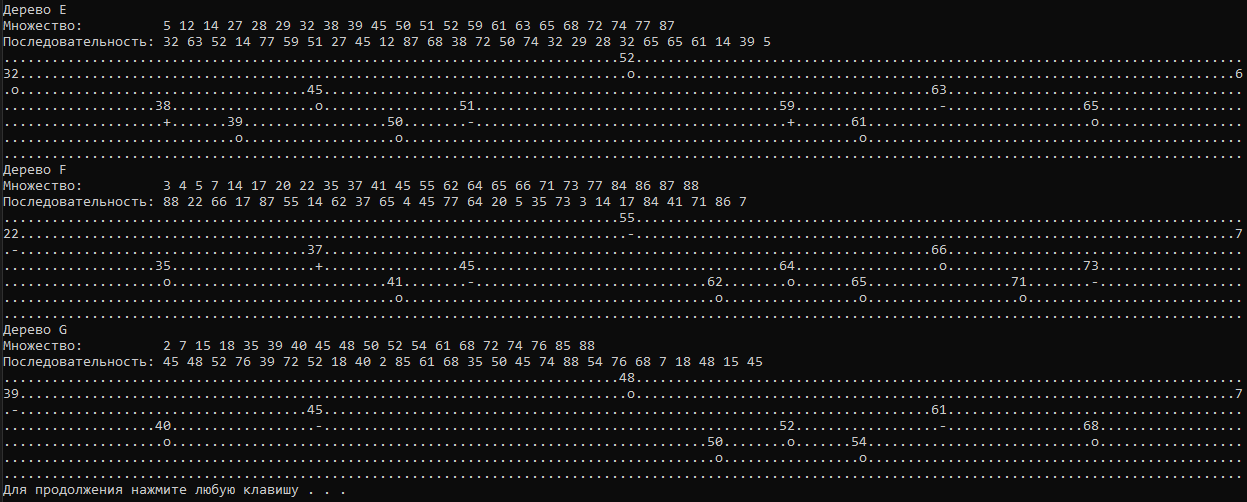


Рис. 2. Деревья E, F, G.

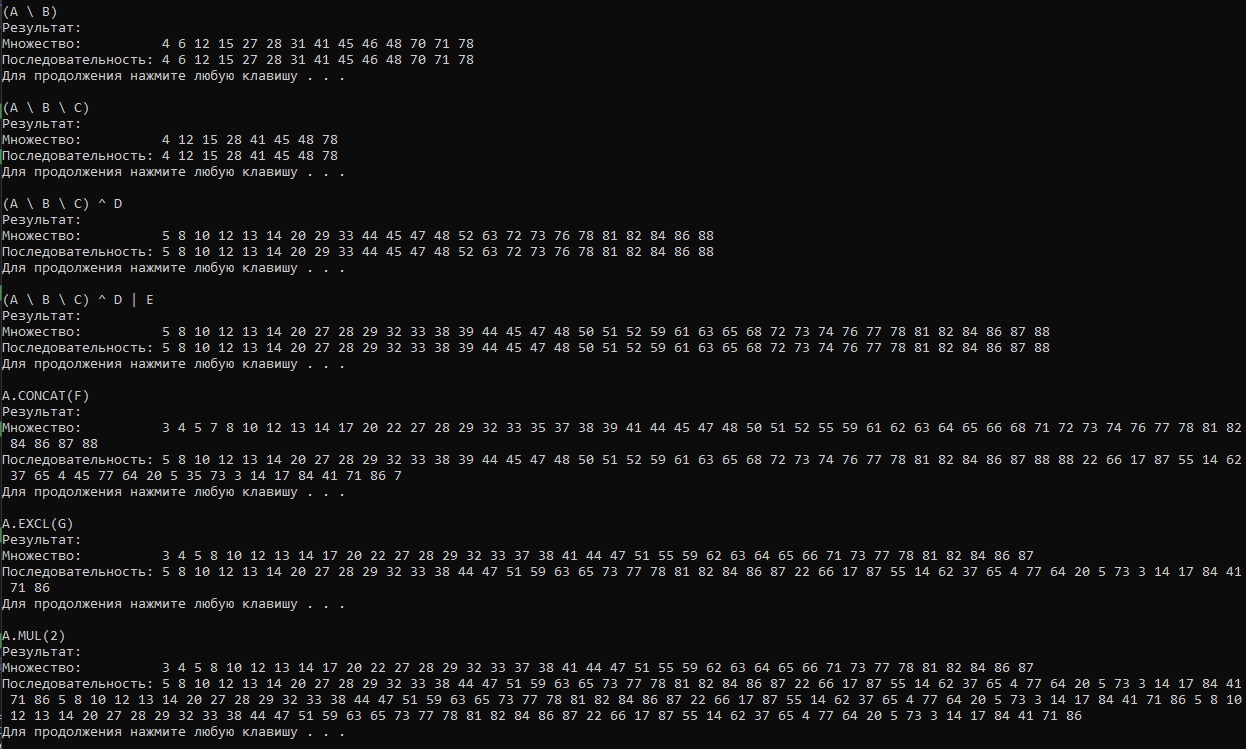


Рис. 3. Цепочка операций (A \ B \ C) ⊕ D ∪ E и операции CONCAT, EXCL, MUL.

**Сравнение хэш-таблицы и ДДП**

- в хэш-таблице поиск, вставка и удаление имеют сложность О(1), в ДДП О(log n);

- в хэш-таблице выделяется память «с избытком», в ДДП ровно на количество элементов;

- в хэш-таблице сложно подобрать подходящую хэш-функцию;

- при сильном расширении множества, хэш-таблицу придется расширять, что может быть «дорогой» операцией;

- в хэш-таблице могут возникнуть коллизии, в ДДП можно воспользоваться автобалансировкой;

- из ДДП легко получить упорядоченный список.

**Вывод**

При выполнении работы, были получены новые знания в области структур данных и их реализации. Была спроектирован свой пользовательский контейнер и отработаны методы написания операций с множествами и последовательностями. Спроектированный пользовательский контейнер представляет собой АВЛ-дерево (с автобалансировкой). Так же были получены новые знания в сфере возможностей языка С++ и о его библиотеках.

**Список использованной литературы**

1. П.Г. Колинько – «Пользовательские контейнеры» учебно-метод. пособие, 2023 г.
2. Сайт https://habr.com/ru/all/