

Министерство науки и образования РФ
Федеральное государственное автономное образовательное
учреждение высшего профессионального образования
«Санкт-Петербургский государственный электротехнический
университет «ЛЭТИ» им. В. И. Ульянова (Ленина)»
(СПбГЭТУ «ЛЭТИ»)
Факультет компьютерных технологий и информатики

Кафедра вычислительной техники

Отчёт
по лабораторной работе № 3
на тему:
“Последовательности”
по дисциплине “Алгоритмы и структуры данных”
Вариант 19

Выполнил студент гр. 4306: Табаков А.В.
Принял: Колинко П.Г.

Цель

Получить практические навыки работы с операциями над последовательностями в виде хеш-таблиц.

Задание

Составить и отладить программу для выполнения заданных операций над последовательностями.

Уточнение задания

Реализуемые операции:

1. Concat – Сцепление. Вторая последовательность подсоединяется к концу первой, образуя ее продолжение.
2. Erase – Укорачивание. Из последовательности исключается часть с порядковыми номерами из отрезка [p1, p2]
3. Excl – Исключение. Вторая последовательность исключается из первой, если является ее частью.

Структура данных представлена в форме хэш-таблицы, но в отличие от первой лабораторной работы имеет изменения:

1. В элементах списка добавлено дополнительное поле pos, отвечающее за порядковый номер в последовательности.
2. Стали возможны повторы элементов, создающие гарантированные коллизии. Коллизии разрешаются методом цепочек и хранятся в списке.
3. При создании последовательности элементам задаётся порядковый номер.

Контрольные примеры

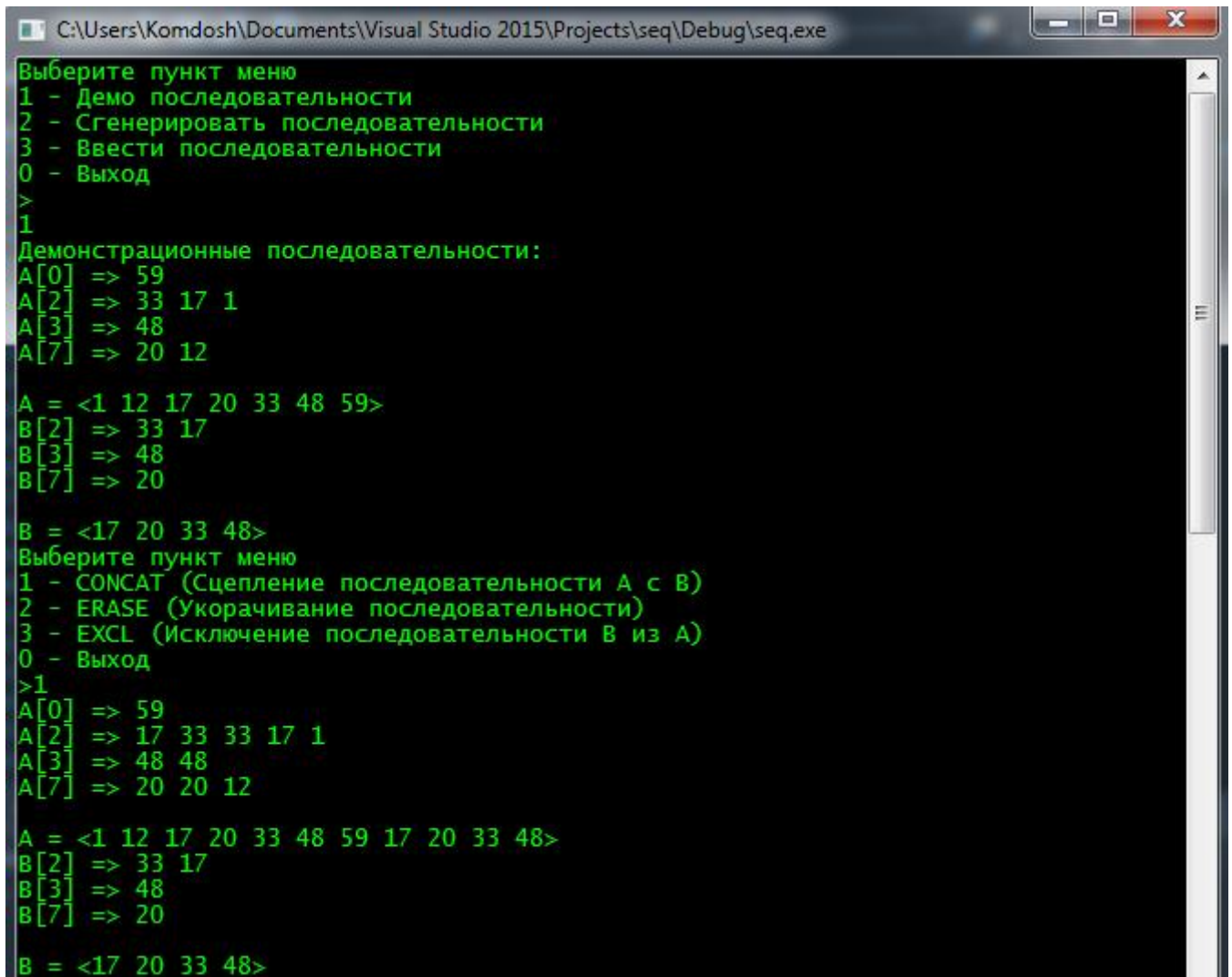
Контрольные примеры представлены в таблице 1.

Таблица. 1. Контрольные примеры

| № | Операция | Исходные множества | | Доп. параметры | Результат |
|---|----------|---------------------|-------------|----------------|---------------------------------|
| | | A | B | | |
| 1 | Concat | 1 12 17 20 33 48 59 | 17 20 33 48 | - | 1 12 17 20 33 48 59 17 20 33 48 |
| 2 | Erase | 1 12 17 20 33 48 59 | 17 20 33 48 | 2 4 0 | 1 12 48 59 |
| 3 | Excl | 1 12 17 20 33 48 59 | 17 20 33 48 | - | 1 12 59 |

Демонстрация программы

1. Демонстрация работы программы с контрольным примером номер 1 из таблицы контрольных примеров.
Код программы см. приложение.



```
C:\Users\Komdosh\Documents\Visual Studio 2015\Projects\seq\Debug\seq.exe
Выберите пункт меню
1 - Демо последовательности
2 - Сгенерировать последовательности
3 - Ввести последовательности
0 - Выход
>
1
Демонстрационные последовательности:
A[0] => 59
A[2] => 33 17 1
A[3] => 48
A[7] => 20 12
A = <1 12 17 20 33 48 59>
B[2] => 33 17
B[3] => 48
B[7] => 20
B = <17 20 33 48>
Выберите пункт меню
1 - CONCAT (Сцепление последовательности A с B)
2 - ERASE (Укорачивание последовательности)
3 - EXCL (Исключение последовательности B из A)
0 - Выход
>1
A[0] => 59
A[2] => 17 33 33 17 1
A[3] => 48 48
A[7] => 20 20 12
A = <1 12 17 20 33 48 59 17 20 33 48>
B[2] => 33 17
B[3] => 48
B[7] => 20
B = <17 20 33 48>
```

2. Демонстрация работы программы с контрольным примером номер 2 из таблицы контрольных примеров.

```
C:\Users\Komdosh\Documents\Visual Studio 2015\Projects\seq\Debug\seq.exe
Выберите пункт меню
1 - Демо последовательности
2 - Сгенерировать последовательности
3 - Ввести последовательности
0 - Выход
>1
Демонстрационные последовательности:
A[0] => 59
A[2] => 33 17 1
A[3] => 48
A[7] => 20 12

A = <1 12 17 20 33 48 59>
B[2] => 33 17
B[3] => 48
B[7] => 20

B = <17 20 33 48>
Выберите пункт меню
1 - CONCAT (Сцепление последовательности A с B)
2 - ERASE (Укорачивание последовательности)
3 - EXCL (Исключение последовательности B из A)
0 - Выход
>2
Введите начальную позицию с которой следует удалить: 2
Введите конечную позицию с которой следует удалить: 4
Введите 0 чтобы укоротить последовательность A, 1 для B: 0
A[0] => 59
A[2] => 33 1
A[3] => 48
A[7] => 20 12

A = <1 12 48 59>
B[2] => 33 17
B[3] => 48
B[7] => 20

B = <17 20 33 48>
```

3. Демонстрация работы программы с контрольным примером номер 2 из таблицы контрольных примеров.

```
C:\Users\Komdosh\Documents\Visual Studio 2015\Projects\seq\Debug\seq.exe
Выберите пункт меню
1 - Демо последовательности
2 - Сгенерировать последовательности
3 - Ввести последовательности
0 - Выход
>1
Демонстрационные последовательности:
A[0] => 59
A[2] => 33 17 1
A[3] => 48
A[7] => 20 12

A = <1 12 17 20 33 48 59>
B[2] => 33 17
B[3] => 48
B[7] => 20

B = <17 20 33 48>
Выберите пункт меню
1 - CONCAT (Сцепление последовательности A с B)
2 - ERASE (Укорачивание последовательности)
3 - EXCL (Исключение последовательности B из A)
0 - Выход
>3
A[0] => 59
A[2] => 33 1
A[3] => 48
A[7] => 20 12

A = <1 12 59>
B[2] => 33 17
B[3] => 48
B[7] => 20

B = <17 20 33 48>
```

Временная сложность

Временная сложность представлена в таблице 2.

Таблица. 2. Временная сложность

| Функция | Средняя | Худшая |
|---------|---------|--------|
| Concat | $O(n)$ | $O(n)$ |
| Erase | $O(n)$ | $O(n)$ |
| Excl | $O(n)$ | $O(n)$ |

Описание алгоритмов

Concat:

Проходим по всем спискам хеш-таблицы левого объекта, цепляя в конец список хеш-таблицы правого объекта, увеличивая позиции элемента на мощность.

Сложить мощность левого объекта и правого объекта, записать в мощность левого объекта.

Erase:

Просматриваем все элементы хеш-таблицы $O(n)$, если позиция элемента больше, чем заданная начальная позиция и меньше чем заданная конечная позиция, то происходит удаление элемента. После просмотра всех элементов снова выполняется цикл просмотра всей хеш-таблицы, для корректировки позиций.

Excl:

Если в объекте слева есть первый по порядку ключ правого объекта, то проходим по всем ключам правого объекта, записывая позиции правых ключей у объекта слева. Если предыдущая позиция ключа левого объекта отличается на иное, чем 1, то выходим. Если цикл завершён, то вызываем функцию erase с указанной начальной позиции, до последней позиции второй последовательности.

Результаты решения задачи

При выполнении программы были получены результаты, совпадающие со значениями, приведенными в таблице 1. Ошибок не обнаружено.

Ответы на контрольные вопросы

1. Во-первых, из-за того, что в ДДП и Хеш-таблицы не хранят порядок ввода, а только сами ключи. Во-вторых, последовательности могут содержать дубликаты ключей и если хеш-таблицы ещё могут поддерживать дубликаты, то сбалансированные деревья не позволяют.
2. Добавление порядкового номера элементам, создание списка последовательности, массив указателей на ключи.
3. Нет, они все будут иметь плюсы и недостатки, например, в список легко вставлять элемент последовательности порядка, но сложно искать элемент.
4. Влияет так как появляются дополнительные поля, которые тоже требуется заполнять.
5. На наш взгляд ДДП, т.к. они не предназначены для хранения дубликатов.
6. Список, можно идти по двум последовательностям одновременно, получив временную сложность $O(n)$.
7. Список, т.к. не потребуется перестройки текущей структуры, но нужно потратиться на нахождение позиции.
8. Массив указателей на ключи, т.к. при замене нужно быстро найти элемент и просто заменить значение.

Вывод

При выполнении лабораторной работы были получены практические навыки работы с операциями над последовательностями в виде хеш-таблиц на языке программирования «C/C++».

Список используемых источников

- Алгоритмы и структуры данных: методические указания к лабораторным работам, практическим занятиям и курсовому проектированию. Ч.2 Вып. 1601 / сост.: П.Г. Колинко. - СПб.: Изд-во СПбГЭТУ "ЛЭТИ", 2016. - 48 с.
- Освой C++ самостоятельно за 21 день. Сиддхартха Рао. 688 стр., с ил.; ISBN 978-5-8459-1825-3; 7 издание.
- <http://stackoverflow.com> – Сайт вопросов и ответов по программированию.
- <http://cyberforum.ru> – Форум программистов и сисадминов.

Приложение

Source.cpp – Код программы

Classes.h – Заголовочный файл программы

A.txt, B.txt – Файлы для демонстрации