

**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**

Студент гр. 0305

\_\_\_\_\_

Солодков Н.С.

Студент гр. 0305

\_\_\_\_\_

Боев И. С.

Преподаватель

\_\_\_\_\_

Колинько П. Г.

Санкт-Петербург

2022

### **Цель работы.**

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

### **Задание (вариант 11)**

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: 1-2-дерево
- Дополнительные операции: CONCAT, SUBST, CHANGE
- Цепочка операций:  $A \mid B \wedge C - D - E$
- Мощность каждого множества: 16

## Контейнер Tree

Основной контейнер представлен классом Tree.

Для реализации операций над множествами были объявлены свои поддерживаемые итераторы, и использованы стандартные функции библиотеки STL: `set_union`, `set_intersection`, `set_difference` и `set_symmetric_difference`.

Поддержка последовательностей реализована хранением в каждом узле вектора порядковых номеров, что позволяет не дублировать сами ключи в дереве. Такая форма организации позволяет восстанавливать последовательность за  $O(n)$ , хотя возможно и является достаточно ёмкой с точки зрения памяти.

Предполагаемая сложность выполнения операций над множествами  $O(\log n)$ . Сложность операций над последовательностями в общем случае представляет из себя  $O(n)$ , так как была выбрана стратегия манипуляции с восстановленными векторами последовательностей и обратным преобразованием их в дерево. Таким образом в общем случае: Восстановление последовательностей:  $O(n) \Rightarrow$  Операции над векторами в общем случае  $O(n) \Rightarrow$  Обратное преобразование в 1-2 дерево  $O(n)$ . Ни одна операция не превышает линейную границу.

## Пример работы программы на случайных данных

```
C:\Users\user\source\repos\3_lab-1_team\Debug\3_lab-1_team.exe
1-2 Tree named A (n=12 height=2)
.....44.....
.....32.....45.....
.....23...42.....
.....27.....
.....19.....
.....0.....
.....+16.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named B (n=13 height=1)
.....34.....
.....46.....
.....17.....40.....49.....
.....4...29.....35.....
.....33...39.....
.....1...19.....
.....28.....
.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named C (n=13 height=2)
.....38.....
.....34.....42.....
.....3...37.....
.....25.....
.....33.....
.....24.....
.....26.....
.....12.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named D (n=14 height=2)
.....43.....
.....10.....46.....
.....4...26.....45.....
.....5.0...3316.....
.....4130.....
.....+.+.
Для продолжения нажмите любую клавишу . . .
1-2 Tree named E (n=12 height=2)
.....29.....
.....4.....40.....
.....0...18.....37.....
.....1...22.....
.....20.....32.....
.....7.....
Для продолжения нажмите любую клавишу . . .
```

Рис. 1. Деревья А, В, С, D, Е.

```

1-2 Tree named F (n=13 height=3)
.....21.....
.....9.....39.....
.....6.....17.....38.....49.....
.....4.....12.....26.....46.....
.....1.....47.....
.....1.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named G (n=15 height=1)
.....18.....
.....14.....31.....
.....15.....
.....9.....19.....41.....
.....10.....
.....8.....4637.....
.....7.....45.....
.....+.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named H (n=14 height=1)
.....44.....
.....46.....
.....26.....45.....
.....41.....
.....23.40.43.....
.....8.38.....
.....17.....
.....95.29.....
Для продолжения нажмите любую клавишу . . .
1-2 Tree named I (n=13 height=2)
.....25.....
.....44.....
.....19.....42.....46.....
.....22.....
.....13.....34.....
.....16.....
.....12.....32.....
.....7.....
.....11.....
Для продолжения нажмите любую клавишу . . .

```

Рис. 2. Деревья F, G, H, I.

```

-----A-----
A as a Set: {0 6 7 15 16 19 23 27 32 42 44 45 }
A as a Subsequence: <44 23 32 19 0 16 45 0 42 7 42 42 7 27 6 15 >
Для продолжения нажмите любую клавишу . . .
B as a Set: {1 4 17 19 28 29 33 34 35 39 40 46 49 }
B as a Subsequence: <40 29 34 35 29 29 46 49 39 4 17 19 28 33 1 35 >

-----A|B-----
A as a Set: {0 1 4 6 7 15 16 17 19 23 27 28 29 32 33 34 35 39 40 42 44 45 46 49 }
A as a Subsequence: <0 1 4 6 7 15 16 17 19 23 27 28 29 32 33 34 35 39 40 42 44 45 46 49 >
Для продолжения нажмите любую клавишу . . .
C as a Set: {3 6 11 12 21 24 25 26 33 34 37 38 42 }
C as a Subsequence: <38 3 42 34 21 11 3 25 37 25 6 34 33 12 24 26 >

-----A|B^C-----
A as a Set: {0 1 3 4 7 11 12 15 16 17 19 21 23 24 25 26 27 28 29 32 35 37 38 39 40 44 45 46 49 }
A as a Subsequence: <0 1 3 4 7 11 12 15 16 17 19 21 23 24 25 26 27 28 29 32 35 37 38 39 40 44 45 46 49 >
Для продолжения нажмите любую клавишу . . .

Prepared intersection between A and D
-----
A as a Set: {0 1 3 4 7 11 12 15 16 17 19 21 23 24 25 26 27 28 29 32 35 37 38 39 40 44 45 46 49 }
A as a Subsequence: <0 1 3 4 7 11 12 15 16 17 19 21 23 24 25 26 27 28 29 32 35 37 38 39 40 44 45 46 49 24 17 24 >
D as a Set: {0 4 5 10 16 17 24 26 28 30 33 40 41 43 45 46 }
D as a Subsequence: <26 10 16 26 46 0 41 30 43 30 4 28 33 45 40 5 24 17 24 >
Для продолжения нажмите любую клавишу . . .

-----A|B^C\D-----
A as a Set: {1 3 7 11 12 15 19 21 23 25 27 29 32 35 37 38 39 44 49 }
A as a Subsequence: <1 3 7 11 12 15 19 21 23 25 27 29 32 35 37 38 39 44 49 >
Для продолжения нажмите любую клавишу . . .

Prepared intersection between A and E
-----
A as a Set: {1 3 7 11 12 15 19 21 23 25 27 29 32 35 37 38 39 44 47 49 }
A as a Subsequence: <1 3 7 11 12 15 19 21 23 25 27 29 32 35 37 38 39 44 49 47 1 >
E as a Set: {0 1 4 7 9 18 20 22 29 32 37 40 47 }
E as a Subsequence: <29 0 4 40 40 18 22 9 22 18 1 20 7 37 22 32 47 1 >
Для продолжения нажмите любую клавишу . . .

-----A|B^C\D\E-----
A as a Set: {3 11 12 15 19 21 23 25 27 35 38 39 44 49 }
A as a Subsequence: <3 11 12 15 19 21 23 25 27 35 38 39 44 49 >
Для продолжения нажмите любую клавишу . . .

```

Рис. 3. Цепочка операций  $A | B \wedge C - D - E$ .

```

F as a Set: {1 4 6 9 12 17 21 26 38 39 46 47 49 }
F as a Subsequence: <6 9 17 39 49 21 46 38 12 47 12 39 4 1 21 26 >
G as a Set: {7 8 9 10 14 15 18 19 31 37 40 41 43 45 46 }
G as a Subsequence: <46 14 31 37 41 9 10 8 40 45 18 43 19 15 10 7 >

-----F.Subst (G, 4)-----
F as a Set: {1 4 6 7 8 9 10 12 14 15 17 18 19 21 26 31 37 38 39 40 41 43 45 46 47 49 }
F as a Subsequence: <6 9 17 39 46 14 31 37 41 9 10 8 40 45 18 43 19 15 10 7 49 21 46 38 12 47 12 39 4 1 21 26 >
Для продолжения нажмите любую клавишу . . .
H as a Set: {5 8 9 17 23 26 29 38 40 41 43 44 45 46 }
H as a Subsequence: <44 40 46 23 26 8 38 17 41 5 45 17 41 43 29 9 >

-----F.Change (H, 29)-----
F as a Set: {4 5 6 7 8 9 10 12 14 15 17 18 19 21 23 26 29 31 37 38 39 40 41 43 44 45 46 47 49 }
F as a Subsequence: <6 9 17 39 46 14 31 37 41 9 10 8 40 45 18 43 19 15 10 7 49 21 46 38 12 47 12 39 4 44 40 46 23 26 8 38 17 41 5 45 17 41 43 29 9 >
Для продолжения нажмите любую клавишу . . .
I as a Set: {7 11 12 13 16 19 22 25 32 34 42 44 46 }
I as a Subsequence: <19 13 12 25 42 44 46 22 16 7 34 32 11 32 11 12 >

-----F.Concat (I)-----
F as a Set: {4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 21 22 23 25 26 29 31 32 34 37 38 39 40 41 42 43 44 45 46 47 49 }
F as a Subsequence: <6 9 17 39 46 14 31 37 41 9 10 8 40 45 18 43 19 15 10 7 49 21 46 38 12 47 12 39 4 44 40 46 23 26 8 38 17 41 5 45 17 41 43 29 9 19 13 12 25 42 44 46 22 16 7 34 32 11 32 11 12 >
End of test.
Для продолжения нажмите любую клавишу . . .

```

Рис. 4. Дополнительные операции CONCAT, SUBST, CHANGE

## Вывод.

При выполнении работы, были получены новые знания в области структур данных и их реализации. Была спроектирована своя структура данных и отработаны методы написания операций с множествами и последовательностями. В течении работы применялись разные подходы и в итоге были найдены возможности реализации с учётом всех нюансов 1-2 дерева с автобалансировкой. Так же были получены новые знания в сфере возможностей языка C++ и о его библиотеках.