

**Отчёт по лабораторной работе №2**  
**«Деревья двоичного поиска»**  
**по дисциплине**  
**«Алгоритмы и структуры данных»**

**Вариант 18**

*Выполнили студенты*  
*факультета КТИ группы №3305*  
Лоуцкер Алексей и Григорьева Анна

*Проверил старший преподаватель*  
Колинко Павел Георгиевич

## Оглавление

1. Задание.....	3
2. Теория.....	3
3. Тестовый пример.....	4
4. Оценки временной сложности операций.....	5
5. Выводы.....	5
6. Список использованной литературы.....	5
8. Приложение: исходные тексты.....	6

## 1. Задание

Переделать программу, составленную по теме «Хеш-таблицы», под использование деревьев двоичного поиска с автобалансировкой.

Вариант 18:

Дерево: AVL-дерево с автобалансировкой

Мощность множества: 32

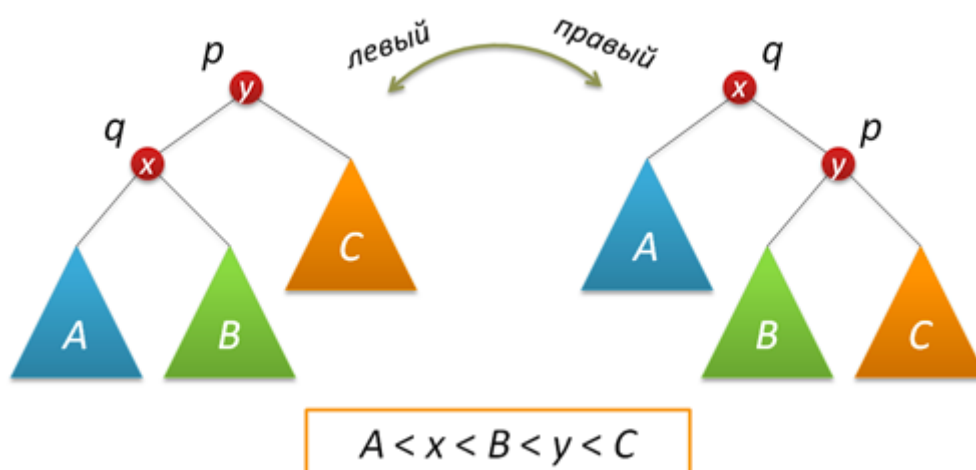
Формула:  $E = A \setminus (B \cap C) \setminus D$

## 2. Теория

AVL-дерево — это прежде всего двоичное дерево поиска, ключи которого удовлетворяют стандартному свойству: ключ любого узла дерева не меньше любого ключа в левом поддереве данного узла и не больше любого ключа в правом поддереве этого узла.

Особенностью AVL-дерева является то, что оно является сбалансированным в следующем смысле: для любого узла дерева высота его правого поддерева отличается от высоты левого поддерева не более чем на единицу.

Узлы хранят не высоту, а разницу высот правого и левого поддеревьев (balance factor), которая может принимать только три значения: -1, 0 и 1. В процессе добавления или удаления узлов в AVL-дерево возможно возникновение ситуации, когда balance factor некоторых узлов оказывается равными 2 или -2, т.е. возникает *расбалансировка* поддерева. Для выправления ситуации применяются повороты вокруг тех или иных узлов дерева.



### 3. Тестовый пример

A:	B:	C:	D:	E:
97	91	92	95	97
94	90	87	93	94
86	83	82	92	86
85	83	80	88	85
83	80	78	80	83
82	77	76	76	82
65	65	68	68	65
60	69	60	66	65
57	76	73	75	57
54	64	52	64	54
52	61	52	60	52
52	58	49	54	52
50	60	52	52	50
49	44	48	52	49
47	43	46	49	47
46	38	43	43	46
42	35	41	40	42
35	32	33	32	35
32	25	31	31	32
26	23	30	30	26
24	19	27	29	24
22	17	27	26	22
18	14	19	25	18
12	13	15	18	12
12	11	11	17	12
9	9	10	11	9
6	5	9	9	6
0	4	5	0	5

## 4. Оценки временной сложности операций

Функции поворотов и балансировки не содержат ни циклов, ни рекурсии, а значит выполняются за постоянное время, не зависящее от размера AVL-дерева.

Операции вставки и удаления (а также поиска) выполняются за время, пропорциональное высоте дерева, т.к. в процессе выполнения этих операций производится спуск из корня к заданному узлу, и на каждом уровне выполняется некоторое фиксированное число действий. А в силу того, что AVL-дерево является сбалансированным, его высота зависит логарифмически от числа узлов. Таким образом, время выполнения всех трех базовых операций логарифмически зависит от числа узлов дерева.

Двухместные операции получают результат в виде вектора значений, а затем последовательно вставляют элементы в новое дерево

Операция	Сложность
балансировка, повороты	$O(1)$
вставка, удаление, поиск	$O(\log(n))$
пересечение, объединение, вычитание	$O(n \cdot \log(n))$

## 5. Выводы

Реализованная структура данных, также как и хеш-таблица, поддерживает простейшие и двухместные операции над множеством ключей. За счет строгой внутренней организации AVL-дерево позволяет получить выигрыш в скорости при выполнении двухместных операций, но из-за необходимости регулярно выполнять балансировку, вставка, поиск и удаление требуют больше времени.

## 6. Список использованной литературы

1) Алгоритмы и структуры данных: методические указания к лабораторным работам, практическим занятиям и курсовому проектированию. Часть 2 / сост. П. Г. Колинко. - СПб.: Изд-во СПбГЭТУ «ЛЭТИ», 2015

2) лекция 9.02.15

2) <http://stackoverflow.com>

3) <http://cplusplus.com>

4) <http://habrahabr.ru/post/150732/>

## **8. Приложение: исходные тексты**

Исходный код доступен в репозитории по адресу:

`github.com/alout1/avl-tree`