

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. Тестовый пример.....	4
3. Оценки временной сложности операций.....	5
4. Выводы.....	5
5. Список использованной литературы.....	5
6. 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: $E = A \vee (B \wedge C) \setminus D$

2. 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
82	80	78	80	83
65	77	76	76	82
60	76	73	68	65
57	69	60	66	65
54	64	52	64	57
52	61	52	60	52
52	60	49	54	52
50	58	52	52	50
49	55	48	52	49
47	44	48	49	47
46	43	46	43	46
42	42	41	40	42
35	38	43	31	35
32	35	39	32	26
26	32	33	30	26
24	25	31	29	24
22	23	30	26	22
18	19	27	25	18
12	20	19	18	12
12	17	15	17	9
9	14	11	11	6
6	11	10	9	5
	9	5	0	
	4			
	0			

3. 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	$O(n \cdot \log(n))$

4. 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.

5. 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/>

6. Приложение: исходные тексты

Исходный код доступен в репозитории по адресу:

`github.com/alout1/avl-tree`