МИНОБРНАУКИ РФ

ФЕДЕРАЛЬНОЕ ГОСУДАРСТВЕННОЕ БЮДЖЕТНОЕ ОБРАЗОВАТЕЛЬНОЕ УЧРЕЖДЕНИЕ ВЫСШЕГО ОБРАЗОВАНИЯ

«ВОСТОЧНО-СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ УНИВЕРСИТЕТ ТЕХНОЛОГИЙ И УПРАВЛЕНИЯ»

(ФГБОУ ВО ВСГУТУ)

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра систем информатики

**Курсовой проект**

**по дисциплине: «Структуры и алгоритмы компьютерной обработки данных»**

**Тема: «Управление сбалансированными бинарными деревьями»**

|  |  |
| --- | --- |
| Выполнил: | студент гр.615 |
| \_\_\_\_\_\_\_\_\_\_\_ | Бадаев М.В. |
| Руководитель: | к.т.н., доцент кафедры СИ |
| \_\_\_\_\_\_\_\_\_\_\_ | Бильгаева Л.П. |
| Оценка: | \_\_\_\_\_\_\_\_\_\_\_ |
| Дата защиты: | \_\_\_\_\_\_\_\_\_\_\_ |
|  | |

Улан-Удэ

2017

ВОСТОЧНО-СИБИРСКИЙ ГОСУДАРСТВЕННЫЙ ТЕХНОЛОГИЧЕСКИЙ

УНИВЕРСИТЕТ

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ЭЛЕКТРОТЕХНИЧЕСКИЙ ФАКУЛЬТЕТ

Кафедра систем информатики

**ЗАДАНИЕ**

на курсовой проект

|  |
| --- |
| **Дисциплина:** Структуры и алгоритмы компьютерной обработки данных |
| **Тема:** Управление сбалансированными бинарными деревьями  **Триссетриса** |
|  |
| **Исполнитель:** Бадаев М.В. |
| **Руководитель:**  Бильгаева Л.П. |
| **Краткое содержание проекта:** Разработка и программная реализация алгоритмов |
| управления сбалансированными бинарными деревьями: поиск, вставка, удаление |
| элементов, балансировка дерева. |
| **1. Теоретическая часть:** АВЛ-дерево. Алгоритмы поиска, включения, удаления |
| элемента. Балансировка дерева. Вращения. |
|  |
|  |
| **2. Практическая часть:** Программная реализация алгоритмов управления, |
| сбалансированными бинарными деревьями |
|  |
| **Сроки выполнения проекта по графику:** |
| 1. Теоретический раздел - 30% к 8 неделе. |
| 2. Основной раздел. Проектирование - 50% к 9 неделе. |
| 3. Основной раздел. Кодирование - 70% к 11 неделе. |
| 4. Экспериментальный раздел - 80% к 12 неделе. |
| 5. Защита - 100% к 13 неделе. |
|  |
| **Требования к оформлению:** |
| 1. Расчетно-пояснительная записка курсового проекта должна быть представлена в электронной и твердой |
| электронной и твердой копиях |
| 2. Объем РПЗ должен быть не менее 20 машинописных страниц без учета приложений |
| 3. РПЗ оформляется по ГОСТу 7.32-91 и подписывается у ответств. за нормоконтроль. нормоконтроль.ннонормоконтроль. |

Руководитель проекта \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Исполнитель \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Дата выдачи " 25 " января \_\_\_\_\_2017 г.

**Аннотация**

Курсовой проект посвящен разработке программы управления сбалансированными бинарными деревьями. Проект направлен на закрепление знаний, полученных в дисциплине «Структуры и алгоритмы обработки данных». В работе программно реализовано визуальное построение сбалансированного двоичного дерева, предусматривающее выполнение таких операций, как поиск, добавление и удаление элемента, а также балансировка дерева.

**Введение**

Целью данного курсового проекта является разработка и реализация программы управления двоичными сбалансированными деревьями на языке высокого уровня, а также закрепление знаний, полученных из курса «Структуры и алгоритмы компьютерной обработки данных».

В программе необходимо было реализовать следующие операции:

1. Поиск элемента в дереве.

2. Добавление элемента в дерево.

3. Удаление элемента из дерева.

4. Балансировка дерева.

Разработанная программа имеет все необходимые средства для наглядного отображения двоичного дерева и выполнения над ним необходимых действий.

Для решения поставленной задачи были выполнены несколько подзадач:

1. Изучение теоретического материала по сбалансированным двоичным деревьям.
2. Изучение интегрированной среды разработки Borland Delphi 7
3. Разработка оптимального алгоритма построения дерева и выполнения над ним требуемых операций.

4. Тестирование разработанного ПО.

**1 ТЕОРЕТИЧЕСКИЙ РАЗДЕЛ**

**1.1 Общая формулировка задания**

Разработать программу на языке высокого уровня, которая позволяет получить наглядное изображение двоичного дерева. Также реализовать основные операции по управлению сбалансированным бинарным деревом:

1. Поиск элемента.
2. Добавление элемента.
3. Удаление элемента.
4. Балансировка дерева.

**1.2 Сбалансированные деревья.**

Одно из определений сбалансированного дерева было предло­жено Г. М. Адельсон-Вельским и Е. М. Ландисом Их критерий сбалансированности сформулиро­ван так:

Дерево называется *сбалансированным* тогда и только тогда, когда высоты двух поддеревьев каждой из его вершин отличаются не более чем на единицу.

Деревья, удовлетворяющие такому условию, часто называют АВЛ-деревьями (по имени их открывате­лей). Так же их просто называют *сбалансирован­ными деревьями,* поскольку упомянутый критерий сбалансированности выглядит наиболее подходящим. (Заметим, что все идеально сбалансированные де­ревья являются также и АВЛ-деревьями.)

Введенное определение не только само очень про­стое, но и приводит к простой процедуре повторной балансировки, причем средняя длина пути поиска практически совпадает с длиной в идеально сбаланси­рованном дереве.

В сбалансированных деревьях за время, пропор­циональное 0(log n) даже в худшем случае, можно выполнить такие операции:

1. Найти вершину с данным ключом.
2. Включить новую вершину с заданным ключом.
3. Исключить вершину с указанным ключом.

Такие утверждения — следствие доказанной Адель­сон-Вельским и Ландисом теоремы, гарантирующей, что сбалансированное дерево никогда не будет по высоте превышать идеально сбалансированное более чем на 45 % независимо от количества вершин. Если обозначить высоту сбалансированного дерева с n вер­шинами через hb(n), то



Оптимум достигается, очевидно, если дерево идеаль­но сбалансировано, при n = 2k — 1.

Фактор баланса узла - высота его правого поддерева минус высота его левого поддерева. Узел с фактором баланса 1, 0, или -1 считают уравновешенным (сбалансированным). Узел с любым другим фактором баланса считают неуравновешенным и требует перебалансировки дерева. Фактор баланса или сохранен непосредственно в каждом узле или вычислен от высот поддеревьев.

**2 Проектный раздел**

**2.1 Алгоритмы основных операций**

Основные операции на АВЛ-дереве вообще предусматривают выполнение тех же самых действий, что и на несбалансированном двоичном дереве поиска, но также сопровождается одной или более операциями, названными *поворотами дерева*, которые помогают восстановить баланс высоты поддеревьев.

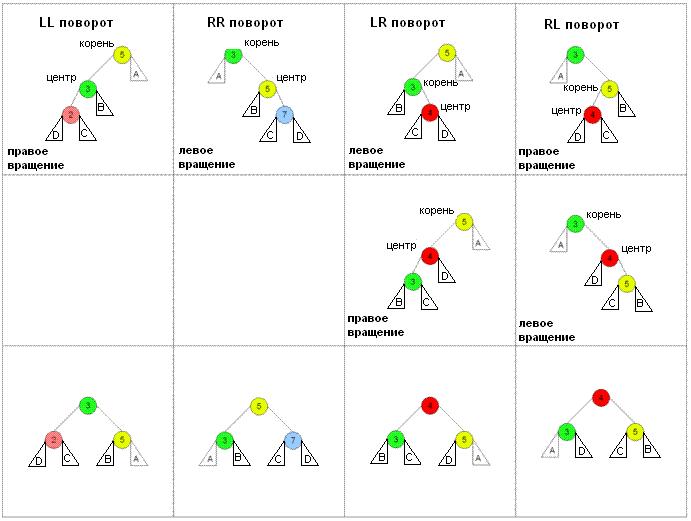


Рис. 1 – Виды поворотов

* 1. **Алгоритм поиска**

Поиск в АВЛ-дереве выполнен точно так же как и в несбалансированном двоичном дереве поиска. Поиск начинается с корневой вершины дерева. Искомый элемент сравнивается с элементом корневой вершины. Если они равны, то элемент найден. Иначе, если искомый элемент меньше корневого, то поиск продолжается по левому поддереву. Если же он больше корневого, то поиск продолжается по правому поддереву. Поиск продолжается до тех пор пока элемент не будет найден, или же если сравниваемый элемент не имеет поддерева, по которому необходимо продолжить поиск. Из-за балансировки дерева поиск выполняется за время O(log n). Если каждый узел дополнительно делает запись размера своего поддерева (включая себя и потомков), то узлы могут быть восстановлены с помощью индекса также за время O (log *n*).

**2.3 Алгоритм вставки**

Вставка элемента в АВЛ-дерево может быть выполнена следующим образом: элемент добавляется в дерево также как будто бы это несбалансированное двоичное дерево поиска, а затем возвращаясь той же дорогой к корню дерева обновляется фактор баланса каждого узла.

Если фактор баланса становится -1, 0, или 1, тогда дерево находится все еще остается сбалансированным и никакие вращения не нужны

Если фактор баланса становится 2 или -2, тогда дерево, корнем которого является этот узел, не неуравновешенно. В этом случае для восстановления баланса необходимо одинарное или двойное вращение.

Существует 4 основных случая вращений, два из которых симметричны другим двум. Обозначим корень несбалансированного дерева P, его правое поддерево – R, левое – L. Если фактор баланса P равен 2, это означает что правое поддерево данного узла перевешивает левое, тогда проверяется фактор баланса правого поддерева (R). Если фактор баланса R равен 1, это означает, что вставка произошла с правой стороны этого узла и необходимо левое вращение с корнем P (RR поворот на рис.1).

Если фактор баланса R равен -1, это означает, что вставка произошла с левой стороны этого узла и необходимо двойное вращение. Первое вращение - правое с корнем R. Второе - левое с корнем P (RL поворот на рис.1).

Другие два случая идентичны предыдущим двум, но с первоначальным фактором баланса -2 и левым поддеревом, перевешивающим правое поддерево (LL и LR повороты на рис.1).

* 1. **Алгоритм удаления**

Если узел - лист, то он просто удаляется. Если узел не лист, он заменяется наибольшим в его левом поддереве и удаляется. Узел, который был найден как замена, имеет самое большее одно поддерево. После удаления восстанавливается связь с элементом - родителем замещенного, также с переопределением факторов баланса.

Восстановление останавливается если фактор баланса становится -1 или 1, указывая что высота этого поддерева осталась неизменной. Если фактор баланса становится 0, тогда высота поддерева уменьшилась на один, и следовательно восстановление должно продолжиться. Если фактор баланса становится -2 или 2 тогда, поддерево несбалансированно и необходимо вращение. Если в результате вращения фактор баланса поддерева остается 0, тогда восстановление к корню должно продолжиться, пока высота этого поддерева не уменьшится на один. В отличие от вставки, где вращение, приводящее к фактору баланса 0, указывает, что высота поддерева осталась неизменной.

**ПРОГРАММНЫЙ РАЗДЕЛ**

* 1. **Интерфейс программы**

Интерфейс программы представляет собой оконное приложение. Логически интерфейс можно разделить на 2 области:

* поле вывода графического представления дерева
* панель функциональных кнопок операций над деревом.

Поле вывода графического представления дерева представляет собой элемент Image, в котором и происходит наглядное отображение дерева. В нем отображаются узлы дерева, линии, отображающие связи между этими узлами, и показатели фактора баланса для каждого узла.

На панели функциональных кнопок расположены элементы интерфейса:

* кнопка «Добавить» - предназначена для добавления элемента в дерево;
* кнопка «Удалить» - предназначена для удаления элемента из дерева;
* кнопка «Скрыть/показать баланс» - предназначена для отображения (скрытия) фактора баланса каждого узла;
* кнопка «Очистить» - предназначена для удаления дерева и очистки экрана;
* кнопка «Выход» - предназначена для завершения работы программы.