Санкт-Петербургский политехнический университет Петра Великого

Институт металлургии, машиностроения и транспорта

Кафедра мехатроники и роботостроения при ЦНИИ РТК

**Курсовая работа**

на тему «Красно-черное дерево»

по дисциплине «Программирование на языках высокого уровня»

Выполнил

Студент гр.33335/2 Мальцева Н.А.

Преподаватель Ананьевский М.С.

<<\_\_\_>>\_\_\_\_\_\_\_\_\_\_2018 г.

Санкт-Петербург

2018

1. **Введение**

Для решения различных задач возникает проблема выбора наиболее подходящей структуры данных. Одной из таких структур являются бинарные деревья. Дерево представляет собой связный граф без петель и циклов. Бинарное дерево – дерево, состоящее из конечного множества элементов, которое содержит корень, связывающий два различных бинарных дерева – левое и правое поддеревья. Каждый элемент дерева – узел, связи между узлами – ветви дерева. Графическая интерпретация бинарного дерева представлена на рисунке 1.

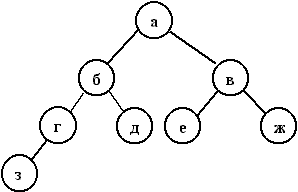


Рисунок 1 – Бинарное дерево

Здесь А – корень дерева, Б – корень левого поддерева, В – корень правого поддерева. Корень дерева расположен на уровне с минимальным значением. Узел Г, который находится непосредственно под узлом Б, называется потомком Б. Узел Б называется родителем или предком узла Г. Каждый узел имеет значение (оно же является и ключом) и ссылки на левого и правого потомка. Если Г находится на уровне i, то Б находится на уровне i-1. Максимальный уровень элемента дерева называется его высотой. Если элемент не имеет потомков, он называется листом дерева. Остальные элементы – внутренние узлы дерева. Число ветвей, которое нужно пройти от корня к узлу x, называется длиной пути к x. Корень имеет длину пути равную 0; узел на уровне i имеет длину пути равную i. Бинарное дерево применяется в тех случаях, когда в каждой точке вычислительного процесса должно быть принято одно из двух возможных решений. Одной из распространенных задач, для которых применяется бинарное дерево, является выполнение заданной операции с каждым элементом дерева или задача обхода дерева.

Бинарное дерево поиска — это бинарное дерево, обладающее дополнительными свойствами: значение левого потомка меньше значения родителя, а значение правого потомка больше значения родителя для каждого узла дерева. То есть, данные в бинарном дереве поиска хранятся в отсортированном виде. При каждой операции вставки нового или удаления существующего узла отсортированный порядок дерева сохраняется. При поиске элемента сравнивается искомое значение с корнем. Если искомое больше корня, то поиск продолжается в правом потомке корня, если меньше, то в левом, если равно, то значение найдено и поиск прекращается.

Сбалансированное бинарное дерево поиска — это бинарное дерево поиска с логарифмической высотой. Более точно оно выражается отношением глубины самого глубокого и самого неглубокого листа (в красно-черных деревьях). В сбалансированном бинарном дереве поиска операции поиска, вставки и удаления выполняются за логарифмическое время (так как дина пути к любому листу от корня не более логарифма). Сбалансированное бинарное дерево поиска применяется, когда необходимо осуществлять быстрый поиск элементов, чередующийся со вставками новых элементов и удалениями существующих. В случае, если набор элементов, хранящийся в структуре данных фиксирован и нет новых вставок и удалений, то массив предпочтительнее. Потому что поиск можно осуществлять алгоритмом бинарного поиска за то же логарифмическое время, но отсутствуют дополнительные издержки по хранению и использованию указателей. Например, в С++ ассоциативные контейнеры set и map представляют собой сбалансированное бинарное дерево поиска.

В данной работе рассматривается красно-чёрное дерево — это одно из самобалансирующихся двоичных деревьев поиска, гарантирующих логарифмический рост высоты дерева от числа узлов и быстро выполняющее основные операции дерева поиска: добавление, удаление и поиск узла. Сбалансированность достигается за счёт введения дополнительного атрибута узла дерева — «цвета». Этот атрибут может принимать одно из двух возможных значений — «чёрный» или «красный». При этом все листья дерева являются фиктивными и не содержат данных, но относятся к дереву и являются чёрными. Графическая интерпретация красно-черного дерева представлена на рисунке 2.

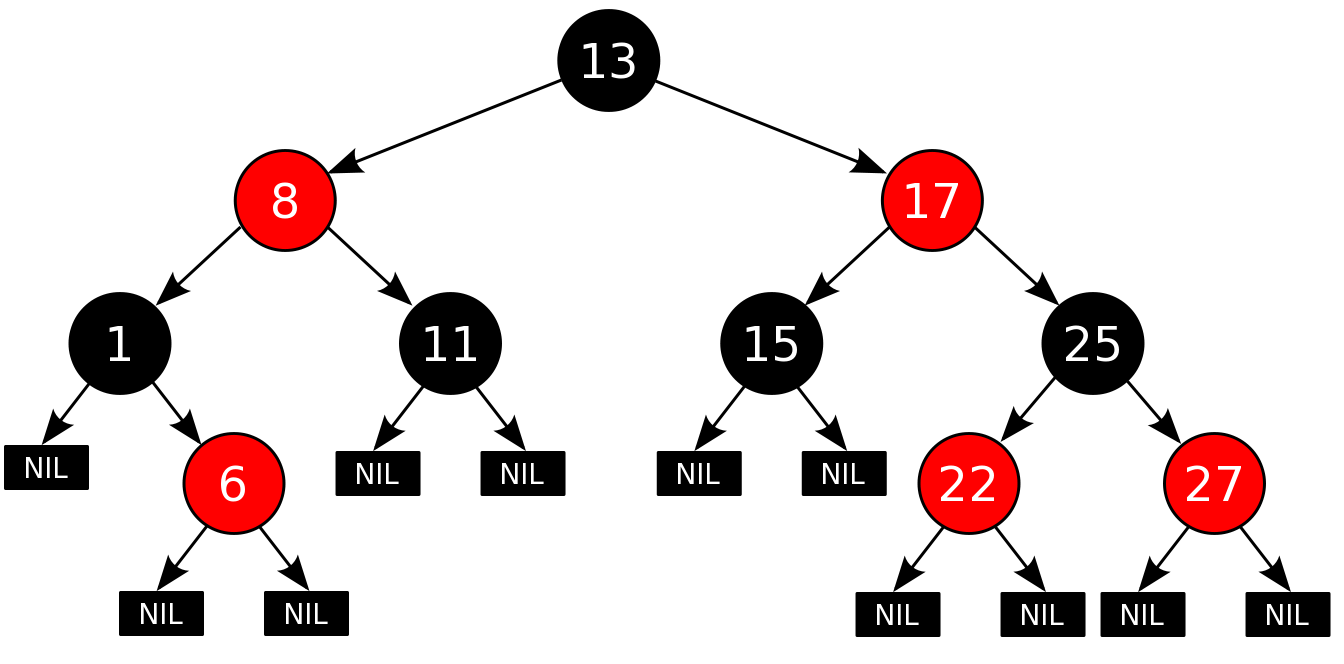


Рисунок 2 – Красно-черное дерево

Красно-черные деревья могут быть использованы для баз данных, в которых хранятся некоторые связанные данные, в которых можно выделить одно из значений (которое более-менее уникально характеризует запись) в качестве ключа. Во многих библиотеках через красно-черные деревья реализуются ассоциативные массивы (абстрактные типы данных, позволяющие хранить пары вида «ключ, значение»).

1. **Постановка задачи**

Изучить и описать словесно алгоритм красно-черного дерева поиска с возможностью добавления, удаления и поиска элементов и его применение. Исследовать эффективность алгоритма по времени работы. Реализовать данный алгоритм на псевдокоде и на языке С или С++.

1. **Описание алгоритма**

Свойства красно-черных деревьев:

1) Каждый узел окрашен либо в красный, либо в черный цвет (в структуре данных узла дополнительное поле – бит цвета).

2) Корень окрашен в черный цвет.

3) Листья (NULL-узлы) окрашены в черный цвет.

4) Каждый красный узел должен иметь два черных дочерних узла. У черного узла могут быть черные дочерние узлы. Красные узлы в качестве дочерних могут иметь только черные.

5) Пути от узла к его листьям должны содержать одинаковое количество черных узлов (это черная высота).