Санкт-Петербургский политехнический университет Петра великого

Институт машиностроения, материалов и транспорта

Высшая школа автоматизации и робототехники

Курсовая проект

Дисциплина: Программирование на языках высокого уровня

Тема: Красно-черное дерево.

Разработал:

студент гр. 3331506/70401 Водорезов Г.И.

Преподаватель Ананьевский М.С.

Санкт-Петербург

2020

Оглавление

[Введение 3](#_Toc41267928)

[Постановка задачи 3](#_Toc41267929)

[Описание алгоритма 3](#_Toc41267930)

[Реализация алгоритма 5](#_Toc41267931)

[Анализ алгоритма 6](#_Toc41267932)

[Сложность алгоритма 6](#_Toc41267933)

[Численный анализ алгоритма 6](#_Toc41267934)

[Область применения 7](#_Toc41267935)

[Заключение 8](#_Toc41267936)

[Список литературы 9](#_Toc41267937)

[Приложение 1 10](#_Toc41267938)

# Введение

## Постановка задачи

Красно-черное дерево - один из видов самобалансирующихся двоичных деревьев поиска. В свою очередь двоичное дерево является решением «словарной проблемы». Подразумевается, что каждой вершине дерева соответствует элемент, имеющий некое ключевое значение, в дальнейшем именуемое просто ключем.

ДДП позволяет выполнять следующие основные операции:

▫ Поиск вершины по ключу.

▫ Переход к предыдущей или последующей вершине, в порядке,

определяемом ключами.

▫ Вставка вершины.

▫ Удаление вершины.

## Описание алгоритма

Эффективность выполнения операций с деревом напрямую связана с его сбалансированностью, то есть с максимальной разницей между глубиной левого и правого поддерева среди всех вершин. Красно-черное дерево является сбалансированным деревом, т.е. деревом, где путь от вершины дерево до любого листа дерева занимает примерно одинаковое время.

Красно-черное дерево обладает всеми свойствами и правилами организации бинарного дерева поиска. Из-за введения дополнительного атрибута узла – цвета, добавляется несколько новых свойств и правил:

1. Узел может быть либо красным, либо чёрным и имеет двух потомков;
2. Корень — как правило чёрный. Это правило слабо влияет на работоспособность модели, так как цвет корня всегда можно изменить с красного на чёрный;
3. Все листья — чёрные и не содержат данных.
4. Оба потомка каждого красного узла — чёрные.
5. Любой простой путь от узла-предка до листового узла-потомка содержит одинаковое число чёрных узлов.

Пример красно-черного дерева представлен на рисунке 1.

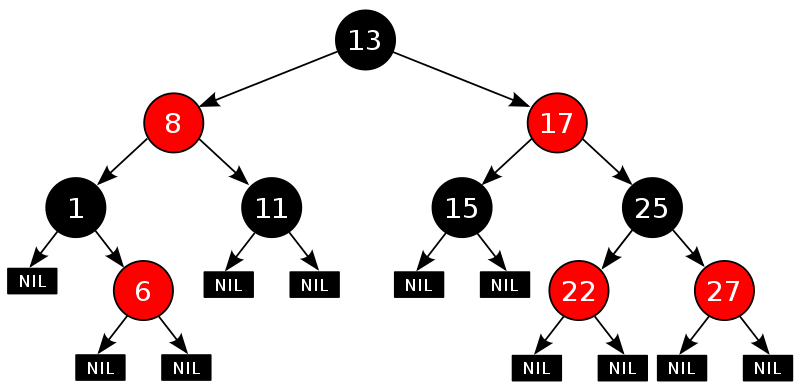


Рисунок 1 – Граф красно-черного дерева

# Реализация алгоритма

Алгоритм был реализован на основе бинарного дерева поиска. Так же существует алгоритм на основе В-дерева. Были реализованы вышеперечисленный функционал двоичного дерева поиска, а также вывод дерева с помощью обхода в ширину и обхода в глубину, замер времени при заполнение разным количеством элементов.

Алгоритм реализован на языке С++, полный код представлен в Приложение 1.

# Анализ алгоритма

## Сложность алгоритма

Так как красно-черное дерево является сбалансированным, то сложность всех операций над узлами одинаковая, так же она одинаковая при худшем и лучшем случае расположения элементов и имеет нотацию O(log N). Занимаемое в памяти место под хранение данных – O(N).

## Численный анализ алгоритма

Посчитаем время заполнения нашего дерева различным количеством данных. Численные результаты приведены в таблице 1.

Таблица 1.

|  |  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| Число элементов, шт. | 101 | 102 | 103 | 104 | 105 | 106 | 107 | 108 | 109 |
| Время выполнения, мс. | 0 | 0 | 0 | 1 | 576 | 2673 | 6320 | 10325 | 16234 |

На рисунке 2 приведен график в логарифмическом масштабе времени выполнения алгоритма в зависимости от количества данных.

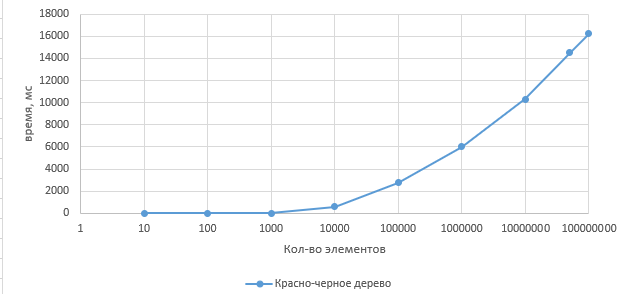


Рисунок 2 – Численный анализ.

# Область применения

Красно-чёрные деревья являются наиболее активно используемыми на практике самобалансирующимися деревьями поиска. В частности, ассоциативные контейнеры библиотеки STL(map, set, multiset, multimap) основаны на красно-чёрных деревьях. TreeMap в Java тоже реализован на основе красно-чёрных деревьев.

# Заключение

В ходе выполнения курсовой работы было рассмотрено красно-черно дерево, выполнена реализация алгоритма на языке С++, проведен анализ сложности алгоритма и его численный анализ. Так же была рассмотрена область применения данного двоичного дерева поиска.

# Список литературы

1. Кормен Т., Лейзерсон Ч., Ривест Р., Штайн К. Алгоритмы: построение и анализ — 2-е изд. — М.: Издательский дом «Вильямс», 2011. — С. 336-364.

# **Приложение** 1

