

МИНИСТЕРСТВО НАУКИ И ВЫСШЕГО ОБРАЗОВАНИЯ РОССИЙСКОЙ ФЕДЕРАЦИИ

Федеральное государственное бюджетное образовательное учреждение

высшего образования

**«МИРЭА – Российский технологический университет»**

**РТУ МИРЭА**



Институт информационных технологий (ИТ)

Кафедра математического обеспечения и стандартизации информационных технологий (МОСИТ)

**ОТЧЕТ ПО ПРАКТИЧЕСКОЙ РАБОТЕ № 8.2**

**по дисциплине**

**«Структуры и алгоритмы обработки данных»**

Тема: «Реализация алгоритмов на основе сокращения числа переборов»

|  |  |  |
| --- | --- | --- |
| Выполнил студент: Лазаренко С.А.  Группа: ИКБО-10-23 |  |  |

Москва 2024

**СОДЕРЖАНИЕ**

[ЦЕЛЬ РАБОТЫ 3](#_Toc178190517)

[ХОД РАБОТЫ 4](#_Toc178190518)

[Формулировка задачи 4](#_Toc178190519)

[Описание подхода к решению 4](#_Toc178190520)

[Коды программы 4](#_Toc178190521)

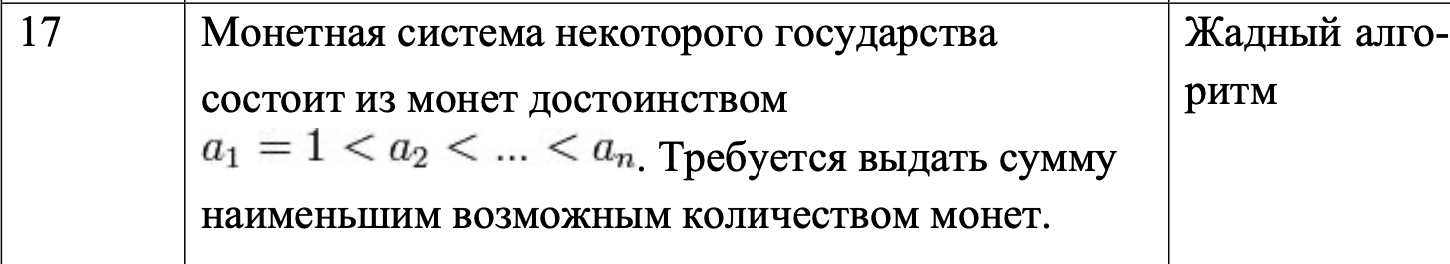
[Результаты тестирования 8](#_Toc178190522)

[ВЫВОД 10](#_Toc178190523)

[СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ 11](#_Toc178190524)

# ЦЕЛЬ РАБОТЫ

Разработать алгоритм решения задачи с применением метода, указанного в варианте и реализовать программу.



Оценить количество переборов при решении задачи стратегией «в лоб» - грубой силы. Сравнить с числом переборов при применении метода. Оформить отчет в соответствии с требованиями документирования.

# ХОД РАБОТЫ

### Формулировка задачи

Начнем с самой крупной монеты, которая меньше или равна сумме SSS, и будем использовать её максимально возможное количество раз. Вычитаем номинал монеты из суммы столько раз, сколько это возможно, и уменьшаем SSS на соответствующее значение. Переходим к следующей по величине монете и повторяем процесс, пока SSS не станет равным нулю. Жадный алгоритм будет работать корректно, если набор монет является каноническим, то есть для любой суммы оптимальное разбиение всегда достигается с использованием наибольших доступных монет.

Индивидуальный вариант работы – 17.

### Описание подхода к решению

### Для реализации красно-черного дерева на C++ будет создан класс, представляющий структуру дерева, в который будут включены методы для вставки узлов и балансировки дерева с использованием правил красно-черного дерева. Каждое вставляемое значение будет добавляться как узел, который будет окрашен в красный цвет по умолчанию. После вставки узла будет выполнена балансировка дерева, которая включает повороты и изменение цветов узлов, чтобы сохранить свойства красно-черного дерева. Для обхода дерева будут реализованы функции для симметричного и прямого обхода, а также методы для подсчета листьев и определения высоты дерева. Визуализация структуры дерева будет осуществляться с помощью рекурсивного вывода, отображающего узлы и их цвета, что позволит пользователю легко понять текущую конфигурацию дерева.

### Код программы

Реализуем код программы на языке программирования C++ :

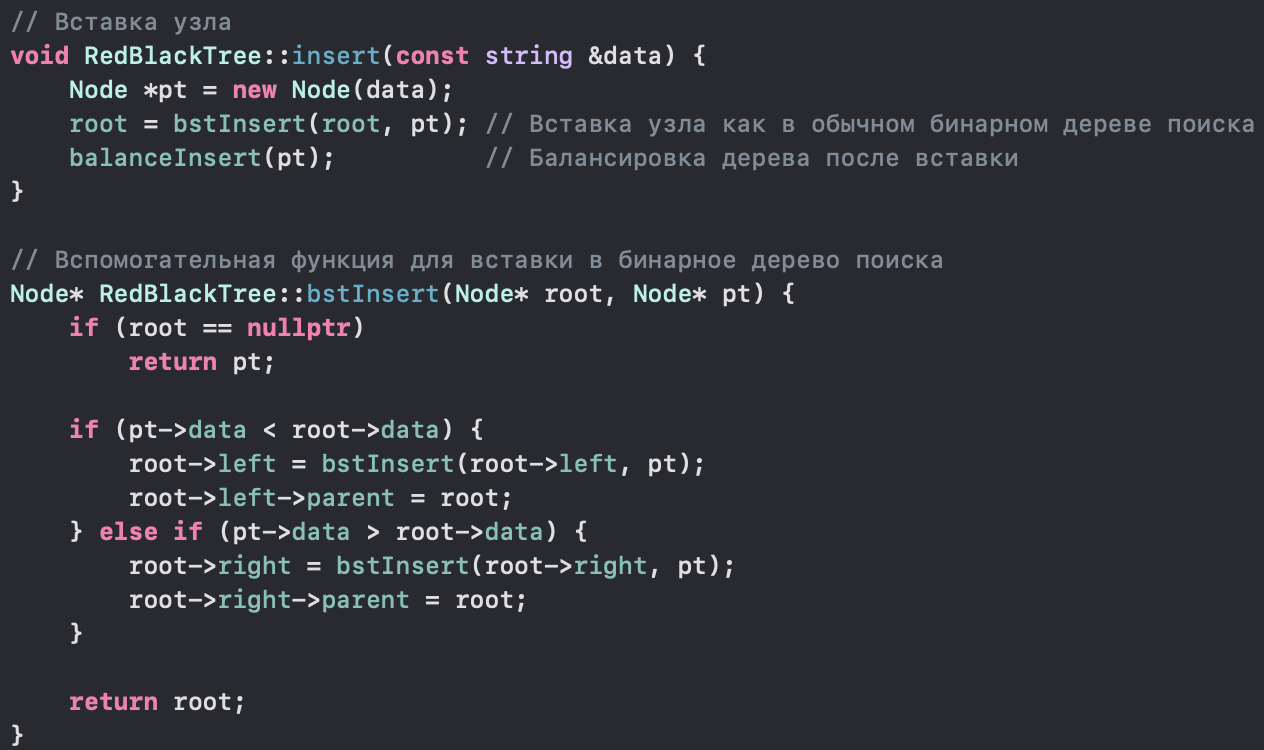


Рисунок 1 – код методов insert и bstInsert

Создает новый узел и вставляет его в дерево, затем вызывает balanceInsert для балансировки.

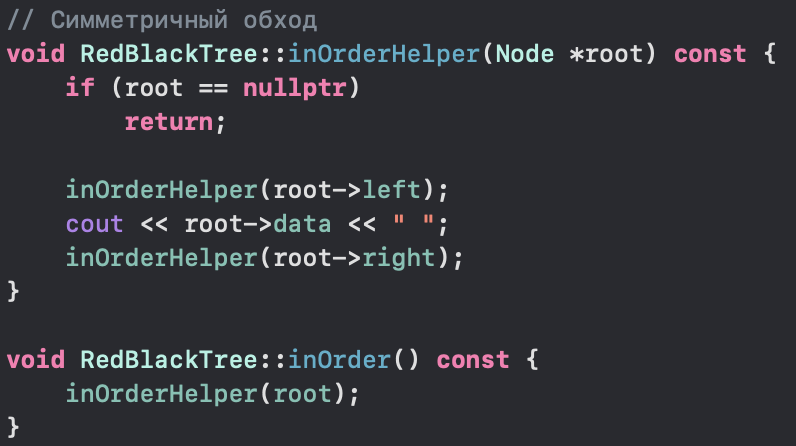


Рисунок 2 – Код методов inOrder и inOrderHelper

Выполняет симметричный обход дерева и выводит узлы.

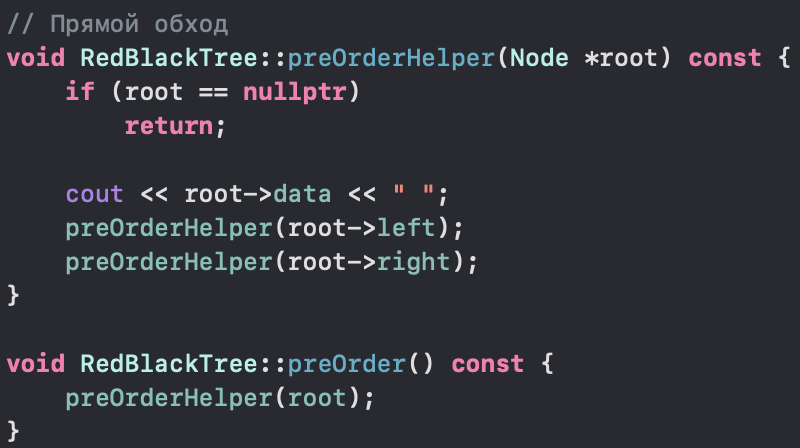


Рисунок 3 – Код методов preOrder и preOrderHelper

Выполняет прямой обход дерева и выводит узлы.

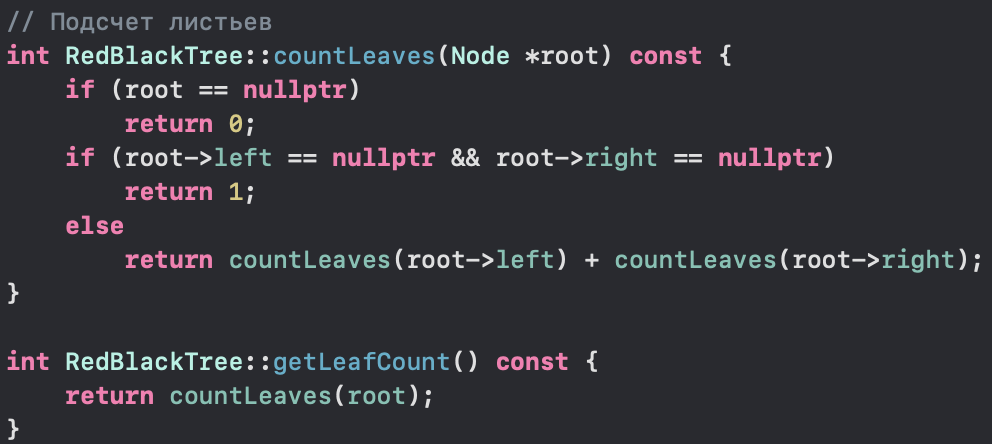


Рисунок 4 – Код метода getLeafCount

Возвращает количество листьев в дереве.

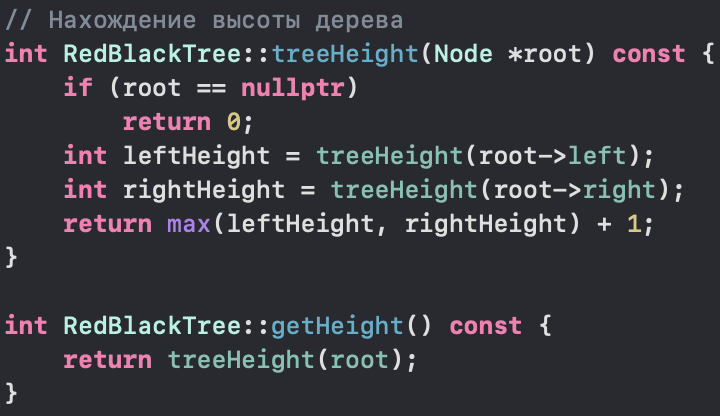


Рисунок 5 – Код метода getLeafCount

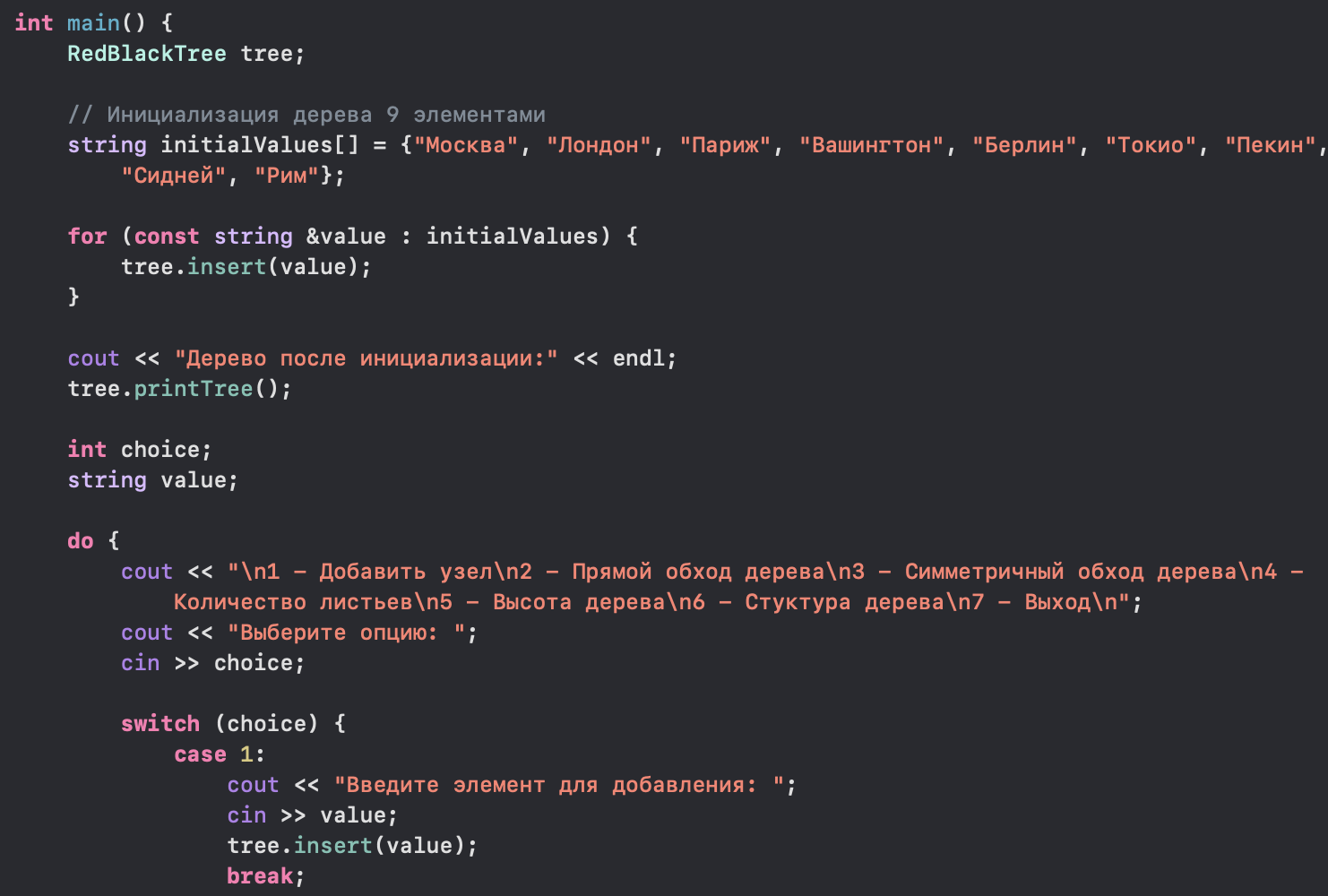


Рисунок 6 – Код main(1 часть)

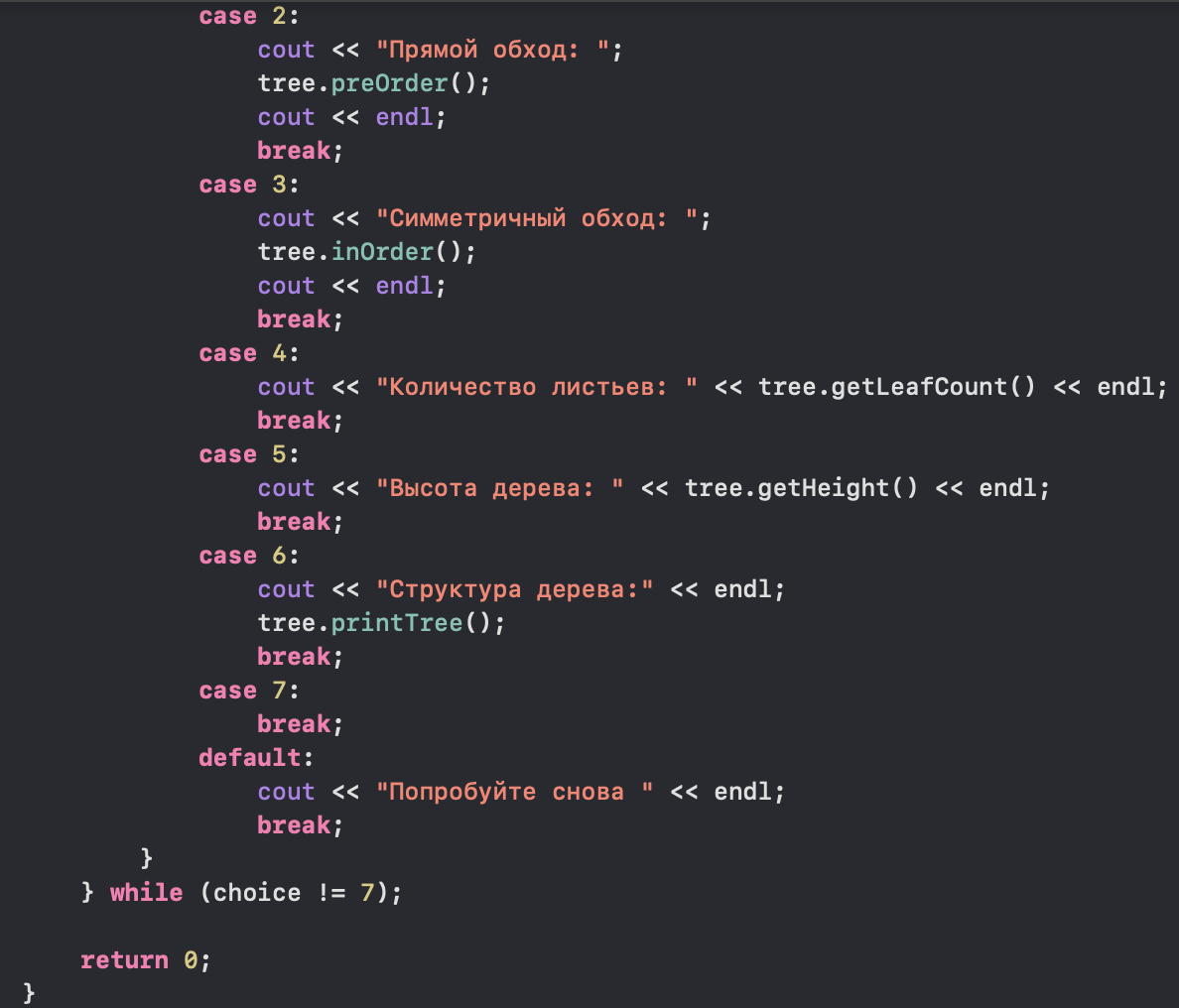


Рисунок 7 – Код main(2 часть)

### Результаты тестирования

Выполним тестирование программы:

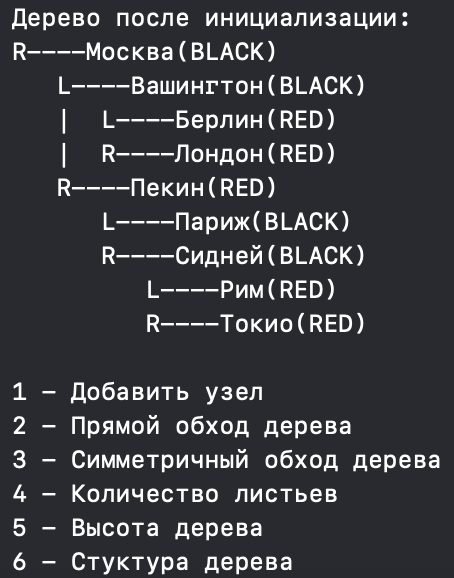


Рисунок 8 – Тестирование программы

Тестирование показало, что программа работает корректно.

# ВЫВОД

Создание класса для красно-черного дерева позволило применять строгие правила цветовой иерархии, что гарантирует, что высота дерева остается логарифмической по отношению к количеству узлов. Благодаря этому обеспечивается высокая производительность при работе с большими объемами данных. Реализованные методы обхода дерева (симметричный и прямой) позволяют эффективно извлекать данные, в то время как функции для подсчета листьев и определения высоты дерева обеспечивают дополнительный анализ структуры. Визуализация дерева помогает пользователю интуитивно понять его конфигурацию и внутреннюю организацию. Таким образом, проект достигает своей цели, предлагая надежное, быстрое и наглядное решение для работы с динамическими наборами данных.

# СПИСОК ИНФОРМАЦИОННЫХ ИСТОЧНИКОВ

1. Страуструп Б. Программирование. Принципы и практика с использованием C++. 2-е изд., 2016.

2. Документация по языку С++ [Электронный ресурс]. URL: https://docs.microsoft.com/ru-ru/cpp/cpp/ (дата обращения 08.09.2024).

3. Курс: Структуры и алгоритмы обработки данных. Часть 2 [Электронный ресурс]. https://online-edu.mirea.ru/course/view.php?id=4020 (дата обращения 04.09.2024)