**МИНОБРНАУКИ РОССИИ**

**САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ**

**«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)**

**Кафедра АПУ**

**ОТЧЕТ**

**по лабораторной работе №4**

**на тему: «Бинарное дерево поиска»**

**по дисциплине «Алгоритмы и структуры данных»**

|  |  |  |
| --- | --- | --- |
| Студент гр. 3391 | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Николаев В.Ю. |
| Преподаватель | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ | Ряскова Е.Б. |

Санкт-Петербург

2024 г.

## Цель работы

## Получить практические навыки реализации и применения алгоритмов для построения и обхода бинарного дерева поиска.

## Задачи

Создать программу, в которой:

* реализован диалог с пользователем для ввода элементов множества, состоящего из целых чисел, и числа элементов этого множества,
* реализован алгоритм построения бинарного дерева поиска на основе заданного множества,
* реализован вывод на экран полученного дерева с использованием прямого, симметричного и обратного вариантов обхода.
* Для взаимодействия с пользователем реализуйте в программе меню, содержащее следующие разделы:
  + Ввод элементов множества
  + Построение и вывод на экран бинарного дерева
  + Выход из программы.

## Описание программы

Программа предоставляет интерактивный интерфейс с меню для выполнения следующих функций:

1. Ввод элементов множества.
2. Построение бинарного дерева поиска и его визуализация с помощью различных методов обхода.
3. Выход из программы.

## Алгоритмы

### Ввод данных:

Функция запрашивает у пользователя количество элементов и сами элементы для построения дерева.

### Построение дерева:

Вставка нового узла в бинарное дерево поиска происходит по следующему алгоритму:

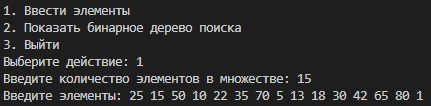
1. **Начало поиска**: поиск начинается с корневого узла.
2. **Сравнение ключей**: сравнивается значение вставляемого ключа с ключом в текущем узле.
   * Если вставляемый ключ меньше текущего ключа, движение продолжается к левому поддереву.
   * Если вставляемый ключ больше, движение направляется к правому поддереву.
3. **Вставка узла**: повторение процесса до достижения пустого места (null ссылки), куда и вставляется новый узел.

### Обходы дерева:

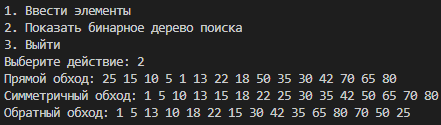
* **Прямой обход** (Pre-order): корень, левое поддерево, правое поддерево.
* **Симметричный обход** (In-order): левое поддерево, корень, правое поддерево.
* **Обратный обход** (Post-order): левое поддерево, правое поддерево, корень.

## Результаты работы программы

Пример ввода данных:



Вывод обходов дерева:



## Выводы

В ходе выполнения лабораторной работы были развиты навыки построения и анализа бинарных деревьев поиска. Были реализованы различные методы обхода деревьев, что позволило более глубоко понять их структуру и особенности работы с такими структурами данных.