

Лабораторная работа 8

Обход бинарного дерева поиска. Поиск максимального и минимального значений

Компьютерная математика II, ММФ, БГУ

Лаврова О.А., май 2025

Навыки

1. **Python**: ООП
2. **математика**: бинарное дерево поиска, центрированный обход

Задание 8.1. Центрированный обход бинарного дерева поиска

1. **Доопределите или переопределите** классы `BinaryTree`, `BinaryNode`, `EmptyNode` из Лабораторной работы 7 за счет определения нового метода `lcr(self)`, реализующего центрированный обход бинарного дерева поиска. Рекомендации по выполнению представлены в лекции к теме *Бинарное дерево поиска. Реализация на основе ООП*.

Результатом вызова метода `lcr` для экземпляра класса `BinaryTree` должен быть список значений вершин дерева, отсортированных по возрастанию.

2. **Создайте** список `source_data` из 10^3 случайных чисел, равномерно распределенных на отрезке $[0, 100]$. **Постройте** бинарное дерево поиска по списку случайных чисел и **выполните** центрированный обход дерева с помощью метода `lcr`.
3. **Изобразите** график по точкам, ординаты которых заданы списком `source_data`. В этой же графической области **изобразите** график по точкам, ординаты которых заданы списком, результатом центрированного обхода бинарного дерева поиска.

Рекомендации по выполнению Задания 8.1

```
In [2]: import matplotlib.pyplot as plt
```

```
In [3]: import random as rnd
```

Создадим список из 10^3 случайных чисел, равномерно распределенных на отрезке $[0, 100]$, с помощью функции `uniform` из модуля `random`

```
In [4]: source_data = [rnd.uniform(0,100) for i in range(10**3)]
```

Построим бинарное дерево поиска на основе списка `source_data`

```
In [5]: tree = BinaryTree()
for i in source_data:
    tree.insert(i)
```

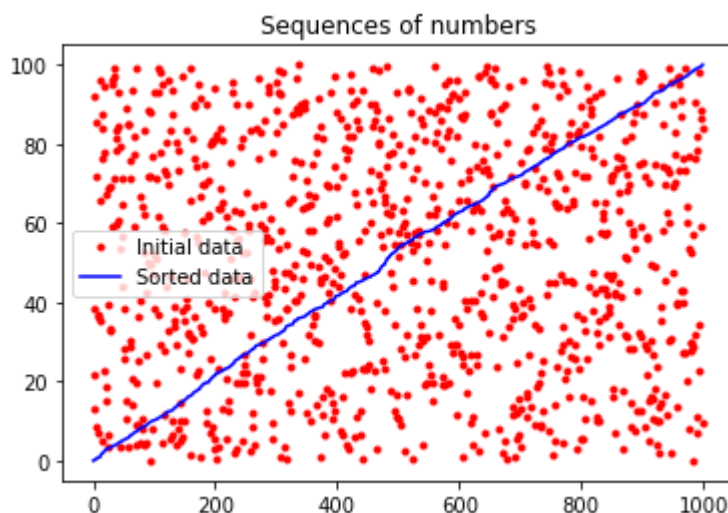
Выполним центрированный обход созданного бинарного дерева `tree`

```
In [6]: sorted_data = tree.lcr()
```

Для проверки корректности обхода изобразим в одной графической области значения исходного списка и результирующего списка после обхода дерева

```
In [7]: plt.plot(source_data, 'r.', label='Initial data')
plt.plot(sorted_data, color='blue', label='Sorted data')
plt.title('Sequences of numbers')
plt.legend()
```

```
Out[7]: <matplotlib.legend.Legend at 0x20b6f409f10>
```



Задание 8.2. Нахождение минимального и максимального значения

Доопределите или переопределите классы `BinaryTree`, `BinaryNode`, `EmptyNode` за счет определения методов `min(self)` и `max(self)` для нахождения минимального и максимального значений в дереве, соответственно.

Рекомендации по выполнению представлены в лекции к теме *Бинарное дерево поиска. Реализация на основе ООП*.

Протестируйте корректность выполнения методов `min` и `max` для дерева `tree`, созданного в Задании 8.1.

Задание 8.3. Создание модуля

Напишите модуль `binaryTree.py` на основе созданных классов.

Напишите строки документации для классов `BinaryTree`, `BinaryNode`, `EmptyNode` и для каждого метода классов.

Протестируйте работу созданного модуля на примерах.

Задание 8.4 Примеры пользовательских классов

Определите три пользовательских класса с использованием возможностей ООП, представленных в лекции к теме "Проектирование классов", например, классовые, статические, абстрактные методы, атрибуты-свойства, шаблон синглтон и др.

Продемонстрируйте на примерах работу трех пользовательских классов.