

# Лабораторная работа 1.

## Самобалансирующиеся деревья поиска



**Даниил Михайлович Берлизов**

Старший преподаватель Кафедры вычислительных систем СибГУТИ

**E-mail:** [sillyhat34@gmail.com](mailto:sillyhat34@gmail.com)

Курс «Структуры и алгоритмы обработки данных»

Осенний семестр, 2021 г.

# Задание на работу, вариант 1

## Вариант 1: красно-чёрные деревья

- Реализовать функции для работы с **красно-чёрными деревьями** (*red-black trees*):

```
struct rbtree *rbtree_add(struct rbtree *root, int key, char *value)
struct rbtree *rbtree_lookup(struct rbtree *root, int key)
struct rbtree *rbtree_delete(struct rbtree *root, int key)
struct rbtree *rbtree_min(struct rbtree *root)
struct rbtree *rbtree_max(struct rbtree *root)
void rbtree_free(struct rbtree *root)
void rbtree_print_dfs(struct rbtree *root, int level)
```

# Задание на работу, вариант 1

- Функцию **rbtree\_delete** реализовать на основе алгоритма, описанного в [CLRS 3ed., с. 356]
- Вставить в красно-чёрное дерево ~50000 элементов, заранее отсортированных в порядке возрастания/убывания
- Используя функцию **rbtree\_lookup**, продемонстрировать логарифмическую зависимость высоты дерева от количества добавленных в него узлов
- Продемонстрировать алгоритм удаления узлов из дерева
- Описать случаи, нарушающие красно-чёрные свойства (при добавлении и удалении узлов)
- Доказать утверждение о высоте красно-чёрного дерева

# Задание на работу, вариант 2

## Вариант 2: АВЛ-деревья

- ♦ Реализовать функции для работы с **АВЛ-деревьями** (*AVL trees*):

```
struct avltree *avltree_add(struct avltree *root, int key, char *value)
struct avltree *avltree_lookup(struct avltree *root, int key)
struct avltree *avltree_delete(struct avltree *root, int key)
struct avltree *avltree_min(struct avltree *root)
struct avltree *avltree_max(struct avltree *root)
void avltree_free(struct avltree *root)
void avltree_print_dfs(struct avltree *root, int level)
```

## Задание на работу, вариант 2

- Функцию **avltree\_delete** реализовать на основе «ленивого» алгоритма (*lazy deletion*)
- Вставить в AVL-дерево ~50000 элементов, заранее отсортированных в порядке возрастания/убывания
- Используя функцию **avltree\_lookup**, продемонстрировать логарифмическую зависимость высоты дерева от количества добавленных в него узлов
- Продемонстрировать алгоритм удаления узлов из дерева
- Описать алгоритмы поворотов AVL-дерева
- Доказать утверждение о высоте AVL-дерева

頑張ってください!