

## Лабораторная работа № 13

# Использование бинарных деревьев

**Задание 13.1.** Найти сумму ключей узлов бинарного дерева поиска (BST), являющихся листьями

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте программу для определения суммы ключей узлов бинарного дерева поиска (BST), являющихся листьями, т.е. не имеющими ветвей.

**Задание 13.2.** Найти сумму квадратов ключей узлов бинарного дерева поиска (BST), имеющих только одну ветвь

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте программу для определения суммы квадратов ключей узлов бинарного дерева поиска (BST), имеющих только одну ветвь, т.е. только правую или только левую.

**Задание 13.3.** Найти произведение ключей узлов бинарного дерева поиска (BST), имеющих ровно две ветви

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте программу для определения произведения ключей узлов бинарного дерева поиска (BST), имеющих ровно две ветви: и правую, и левую.

**Задание 13.4.** Найти наименьший ключ в узлах бинарного дерева поиска (BST), являющихся листьями

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте программу для определения наименьшего ключа среди узлов бинарного дерева поиска (BST), являющихся листьями, т.е. не имеющими ветвей.

**Задание 13.5.** Найти наибольший ключ в узлах бинарного дерева поиска (BST), имеющих только одну ветвь

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте программу для определения наибольшего ключа среди узлов бинарного дерева поиска (BST), имеющих только одну ветвь, т.е. только правую или только левую.

**Задание 13.6.** Найти  $k$ -й по порядку наименьший элемент в бинарном дереве поиска (BST)

Дано бинарное дерево поиска (BST) и натуральное число  $k$ . Разработайте и оттестируйте функцию `k_smallest_element()`, которая находит  $k$ -й наименьший элемент в заданном бинарном дереве поиска.

**Задание 13.7.** Определить медиану ключей бинарного дерева поиска (BST)

Дано бинарное дерево поиска (BST). Разработайте и оттестируйте функцию `findMedian()`, которая возвращает медиану ключей бинарного дерева поиска (BST).

Медиана ключей бинарного дерева поиска (BST) равна (порядок узлов отсортирован по возрастанию ключей):

Если количество узлов в дереве  $N$  четно, то медиана равна среднему арифметическому значению ключей узлов с номерами  $\frac{N}{2}$  и  $\frac{N}{2} + 1$ .

Если количество узлов в дереве  $N$  нечетно, то медиана равна ключу узла с номером  $\frac{N+1}{2}$ .

**Задание 13.8.** Найти ближайший ключ в бинарном дереве поиска (BST)

Дано бинарное дерево поиска (BST) и ключ  $K$  одного из узлов дерева. Разработайте и оттестируйте программу для определения ключа  $K'$  другого узла в дереве, который наиболее близок к ключу  $K$ , т.е. величина  $|K' - K|$  минимальна.

**Задание 13.9.** Преобразовать бинарное дерево поиска (BST) в сбалансированное BST

Дано бинарное дерево поиска (BST), которое, вообще говоря, не является сбалансированным. Разработайте и протестируйте функцию `buildBalancedTree()`, которая конвертирует заданное бинарное дерево поиска в сбалансированное бинарное дерево поиска, которое имеет минимально возможную высоту.

**Задание 13.10.** Найти все пары, нарушающие свойство бинарного дерева поиска (BST)

Дано бинарное дерево, вообще говоря, не являющееся бинарным деревом поиска (BST). Разработайте и протестируйте функцию `pairsViolatingBST()`, которая подсчитывает и выводит на консоль все пары, нарушающие свойство бинарного дерева поиска (BST), а именно, что каждый узел в левом поддереве должен иметь ключ меньше, чем ключ любого узла в правом поддереве.

**Задание 13.11.** Построить оптимальное бинарное дерево поиска (BST)

Дан отсортированный массив ключей поиска `keys[0.. n-1]` и массив частот поиска `freq[0.. n-1]`, где `freq[i]` – частота поиска ключа `keys[i]`. Постройте бинарное дерево поиска из всех ключей таким образом, чтобы суммарная стоимость поиска была минимальной. Под стоимостью поиска понимается сумма стоимостей поиска для узлов дерева, стоимость поиска для узла дерева равна уровню этого узла, умноженному на его частоту. Уровень корневого узла равен 1.

**Задание 13.12.** Распечатать ключи узлов двух бинарных деревьев поиска (BST)

Даны два бинарных дерева поиска (BST). Разработайте и протестируйте программу для печати ключей узлов обоих бинарных деревьев поиска в отсортированном порядке (без повторений), т.е. фактически найдите объединение двух бинарных деревьев поиска.

**Задание 13.13.** Найти парные узлы из двух бинарных деревьев поиска (BST)

Даны два бинарных дерева поиска и заданное натуральное число  $x$ . Разработайте и протестируйте функцию `countPairs()`, которая возвращает все пары из обоих BST, сумма значений которых равна  $x$ . Функция имеет три аргумента на входе: указатель на корневой узел первого дерева, указатель на корневой узел второго дерева и число  $x$ .

**Задание 13.14.** Распечатать ключи одинаковых узлов двух бинарных деревьев поиска (BST)

Даны два бинарных дерева поиска. Разработайте и протестируйте программу для печати ключей одинаковых узлов обоих бинарных деревьев поиска в отсортированном порядке, т.е. фактически найдите пересечение двух бинарных деревьев поиска.

**Задание 13.15.** Преобразовать бинарное дерево в бинарное дерево поиска (BST)

Дано бинарное дерево, вообще говоря, не являющееся бинарным деревом поиска (BST). Разработайте и протестируйте функцию `binaryTreeToBST()`, которая конвертирует бинарное дерево в бинарное дерево поиска и возвращает корневой узел полученного бинарного дерева поиска.

**Задание 13.16.** Проверить, что бинарное дерево является бинарным деревом поиска (BST)

Дано бинарное дерево, вообще говоря, не являющееся бинарным деревом поиска (BST). Разработайте и протестируйте функцию `isBST()`, аргументом которой является корневой узел бинарного дерева, которая возвращает истину, если бинарное дерево является бинарным деревом поиска.

**Задание 13.17.** Определить наибольшее бинарное дерево поиска (BST)

Дано бинарное дерево, вообще говоря, не являющееся бинарным деревом поиска (BST). Разработайте и протестируйте функцию `largestBst()`, которая возвращает размер (количество узлов) наибольшего поддерева, которое является бинарным деревом поиска (BST). Если исходное дерево целиком является бинарным деревом поиска (BST), то вернуть размер всего дерева.