

BPlusTree

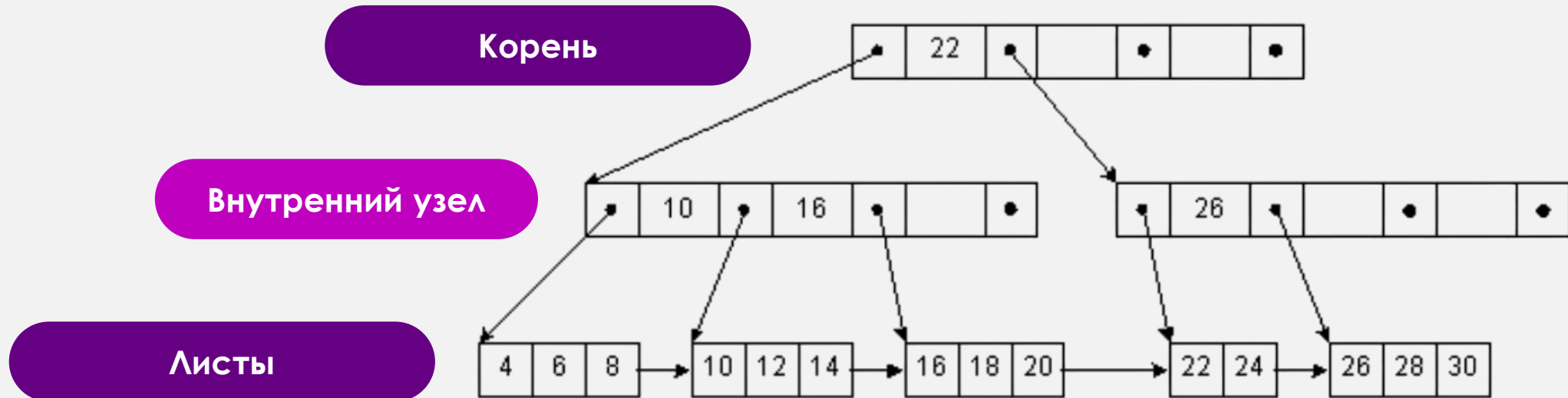
Описание алгоритма

2022 г.



ОПРЕДЕЛЕНИЕ

Как и B-дерево, представляет собой строго сбалансированную древовидную структуру данных, которая поддерживает отсортированные данные и разрешает поиск, последовательный доступ, вставку и удаление в логарифмическом времени.



Все узлы, кроме корня, имеют
как минимум $\lceil m/2 \rceil - 1$ и
максимум $m-1$ ключей

Глубина всех листьев
одинакова

Все ключи хранятся в
порядке возрастания

СВОЙСТВА m -арного Bplus-дерева

Внутренний узел
может иметь
максимум m дочерних
элементов.

Оригиналы записей хранятся
в листовых узлах, внутренние
узлы содержат лишь
значения.

Листы хранятся в виде
односвязного списка

ПОДДЕРЖИВАЕМЫЕ ОПЕРАЦИИ

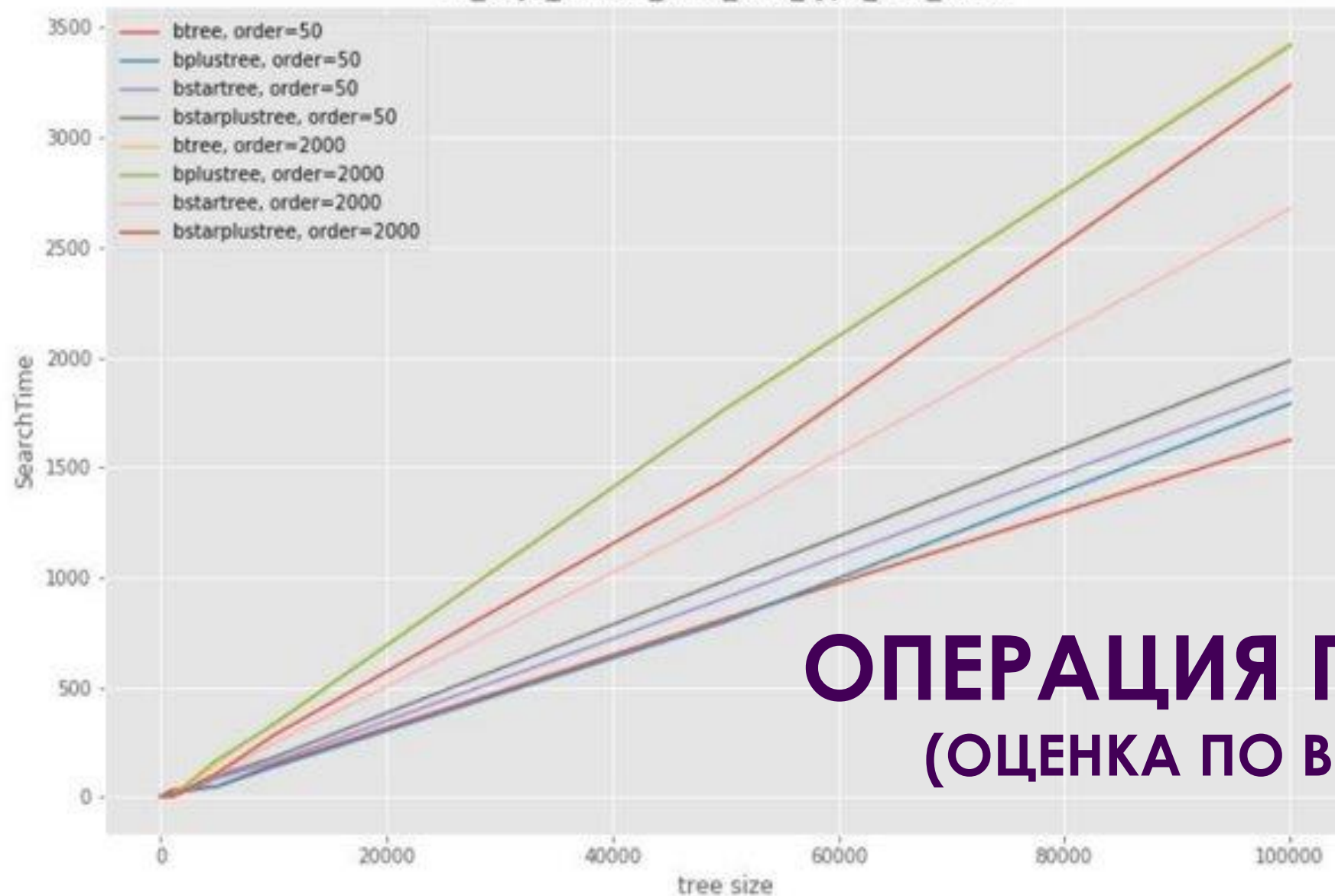
Вставка

Удаление

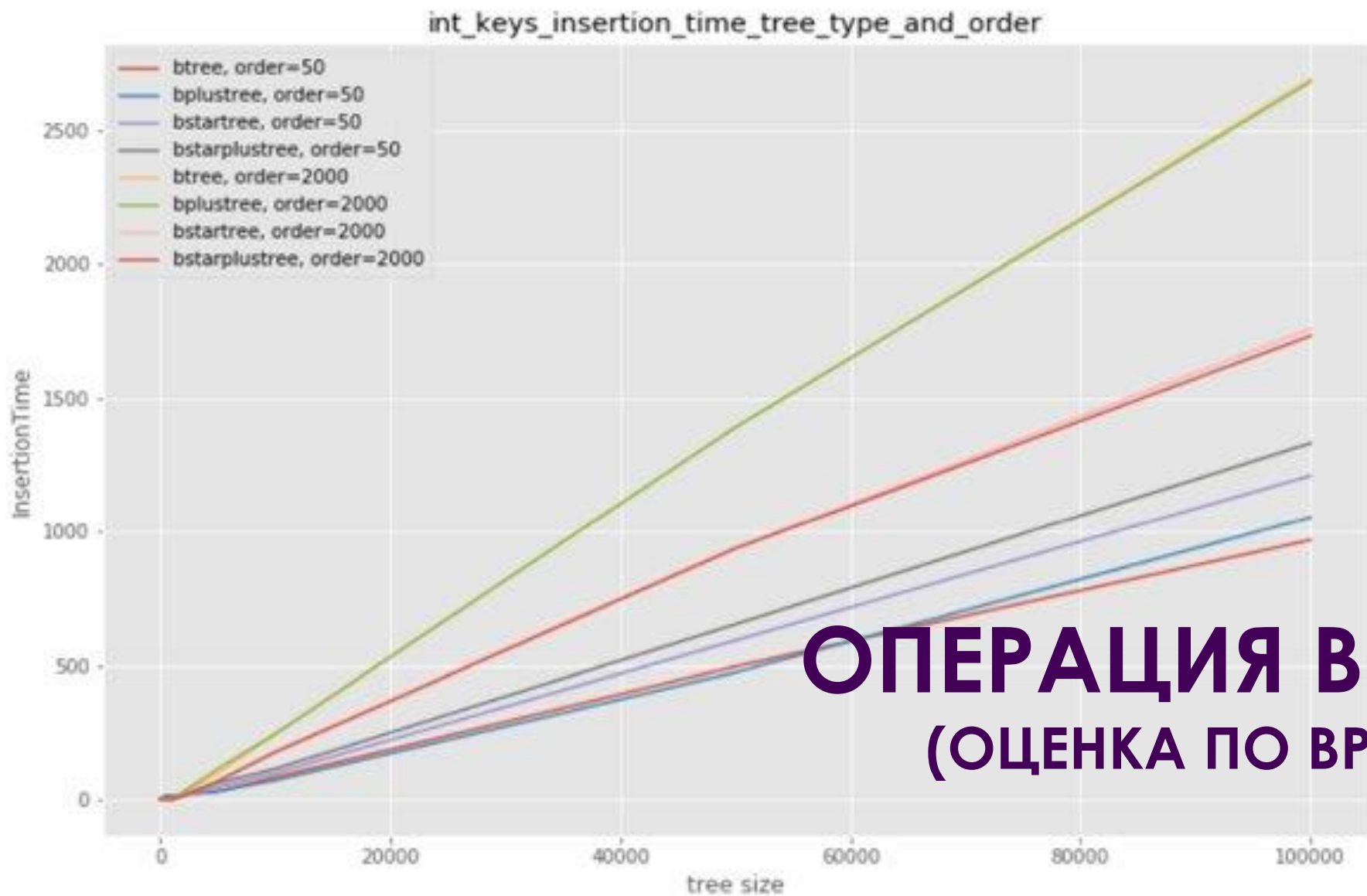
Поиск

Обход (вывод)

int_keys_search_time_tree_type_and_order

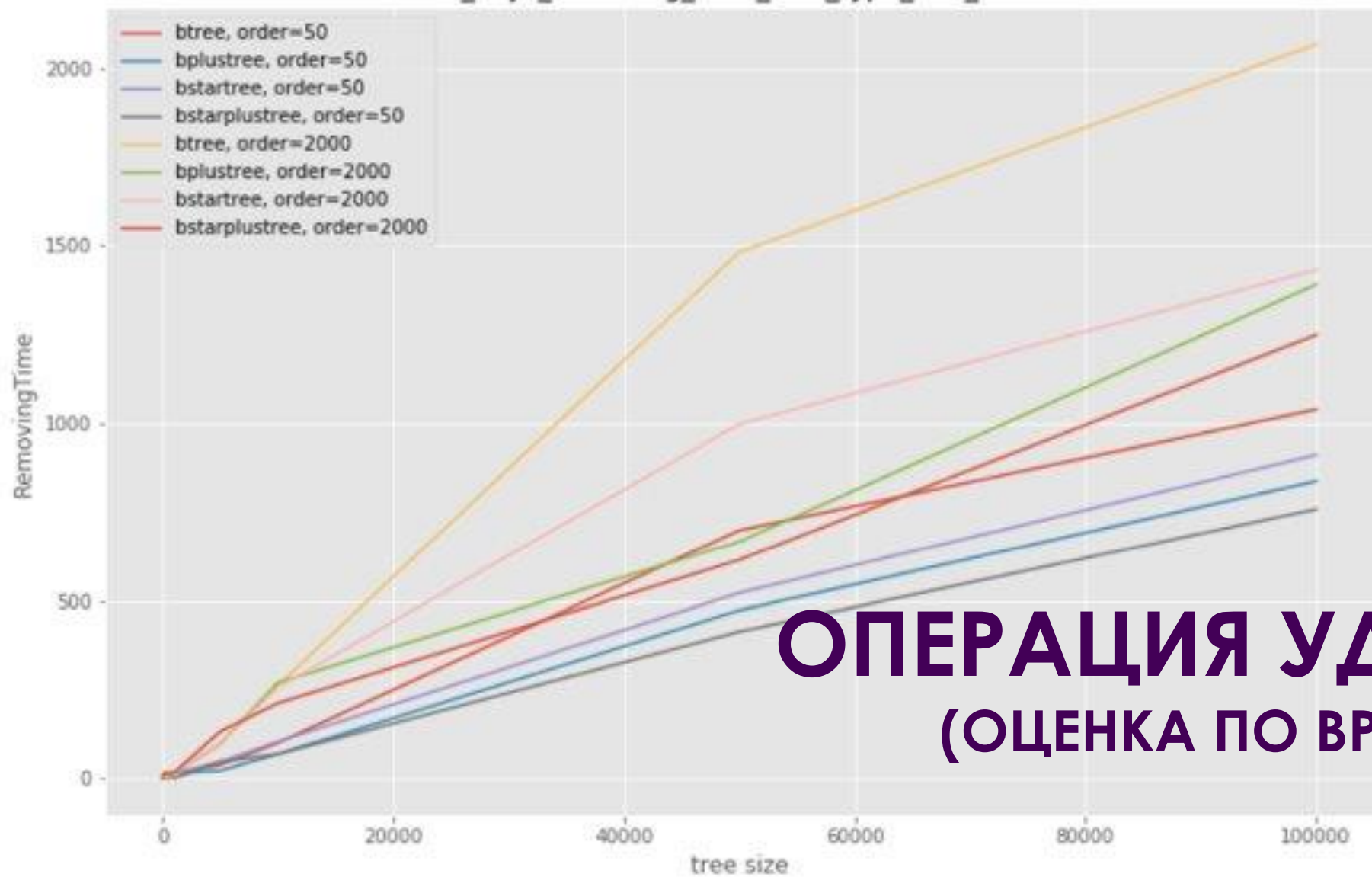


ОПЕРАЦИЯ ПОИСКА
(ОЦЕНКА ПО ВРЕМЕНИ)



ОПЕРАЦИЯ ВСТАВКИ (ОЦЕНКА ПО ВРЕМЕНИ)

int_keys_removing_time_tree_type_and_order



ОПЕРАЦИЯ УДАЛЕНИЯ (ОЦЕНКА ПО ВРЕМЕНИ)

СТРУКТУРА УЗЛА

Массив ключей

Массив дочерних
элементов

Указатель на родителя

Указатель на правого
соседа (если узел
листовой)

СТРУКТУРА ДЕРЕВА

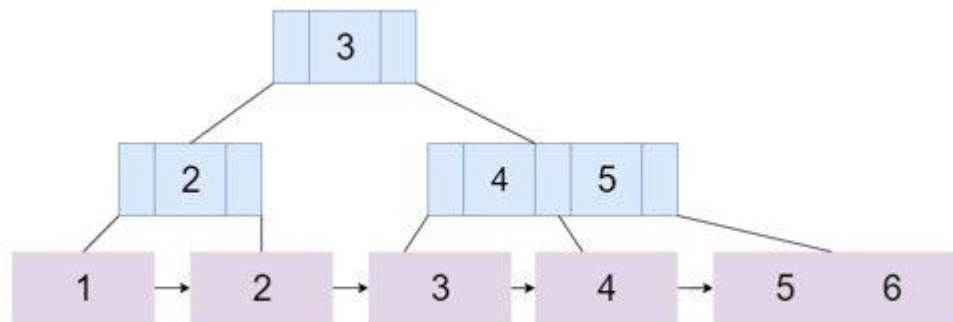
Порядок дерева

Указатель на корень

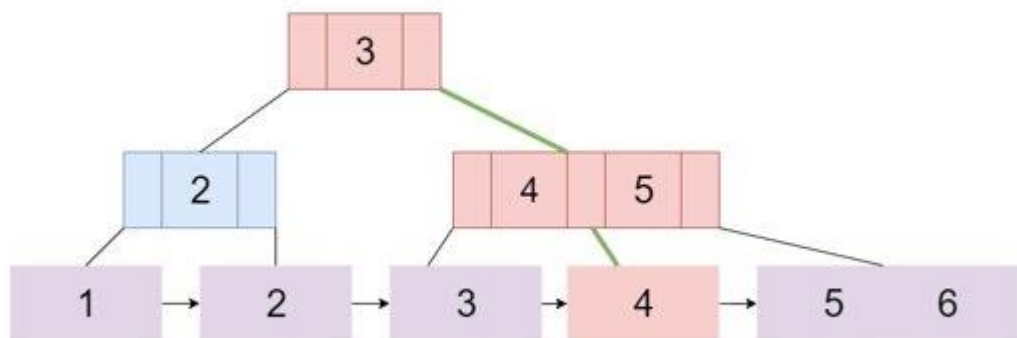
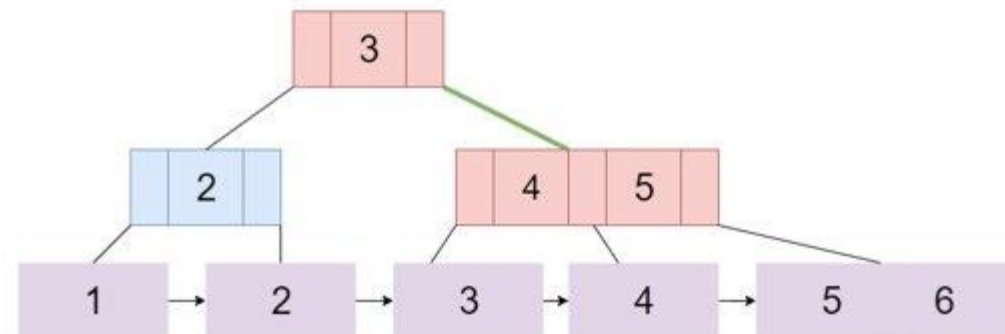
АЛГОРИТМ ПОИСКА

От корневого узла производится поиск интервала содержащий значение ключа и осуществляется переход к соответствующему сыну.

Предыдущий пункт повторяется пока текущий узел не окажется листовым



Поиск k=4



АЛГОРИТМ ДОБАВЛЕНИЯ

Случай 1

Если дерево пустое, нужно создать новый корень и добавить первый ключ. Иначе производится поиск листа, в который можно добавить ключ. Ключ добавляется в список в порядке неубывания. Если лист не заполнен, операция завершена

АЛГОРИТМ ДОБАВЛЕНИЯ

Случай 2

Если после добавления узел оказывается переполненным, происходит разбиение узла. Узел делится на две части, две половины. Элемент $m/2+1$ переносится в родительский узел. Если родительский узел оказывается заполненным, повторяется разбиение узла.

АЛГОРИТМ УДАЛЕНИЯ

Случай 1

Ключ, который нужно удалить, присутствует только в листовом узле, а не во внутренних узлах. Если количество соответствует минимальному, операция завершается. Иначе проводим балансировку.

Балансировка. Срединный ключ потомка добавляется к родителю. Недостающий ключ заимствуется у брата.

АЛГОРИТМ УДАЛЕНИЯ

Случай 2

Ключ, который нужно удалить, присутствует и во внутренних узлах.

А) Если количество ключей в узле больше минимального, ключ просто удаляется из листового и внутреннего узла. Пустое место во внутреннем узле заполняется преемником по порядку.

Б) Если количество ключей в узле меньше минимального, он заполняется ключом заимствованным у брата.

АЛГОРИТМ УДАЛЕНИЯ

Случай 3

Ключ, который нужно удалить, присутствует и во внутренних узлах, но позаимствовать ключ у брат невозможно, т.к. тогда количество элементов в них будет меньше минимального.

Создается пустое пространство, ссылки на которое в последующем удаляются. Далее продельвается слияние корня и ближайшего незаполненного ребенка. Создаются новые ссылки, старые удаляются

**Спасибо за
внимание!**