

B-дерево. B+-дерево.
B*-дерево.

Цель и задачи

- Цель: разработать и описать реализацию алгоритмов B-tree, B+-tree и B*-tree. Протестировать данные алгоритмы с помощью автоматических тестов и сравнить результаты.
- Задачи:
 1. Изучить теоретический материал по B-tree, B+-tree и B*-tree
 2. Описать B-tree, B+-tree и B*-tree
 3. Реализовать B-tree
 4. Продумать тесты, подходящие для каждого вида дерева
 5. Реализовать тесты к B-tree
 6. Реализовать B+-tree
 7. Реализовать B*-tree
 8. Описать результаты тестирования B-tree, B+-tree и B*-tree

Основные операции

$\text{insert}(x)$ – вставить элемент x , если его ещё не существует в дереве;

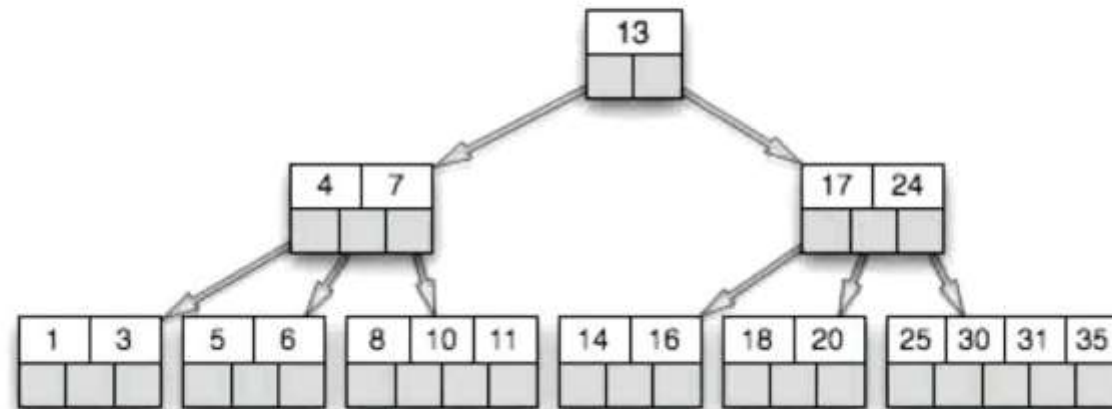
$\text{delete}(x)$ – удалить элемент x , если он есть в дереве;

$\text{search}(x)$ – найти элемент x , если он существует в дереве.

В-дерево

- В-tree — структура данных, сбалансированное, сильно ветвистое дерево поиска, предложенное Р. Бэйером и Э. МакКрейтом в 1970 году для хранения данных во внешней памяти.

B-Tree



Преимущества

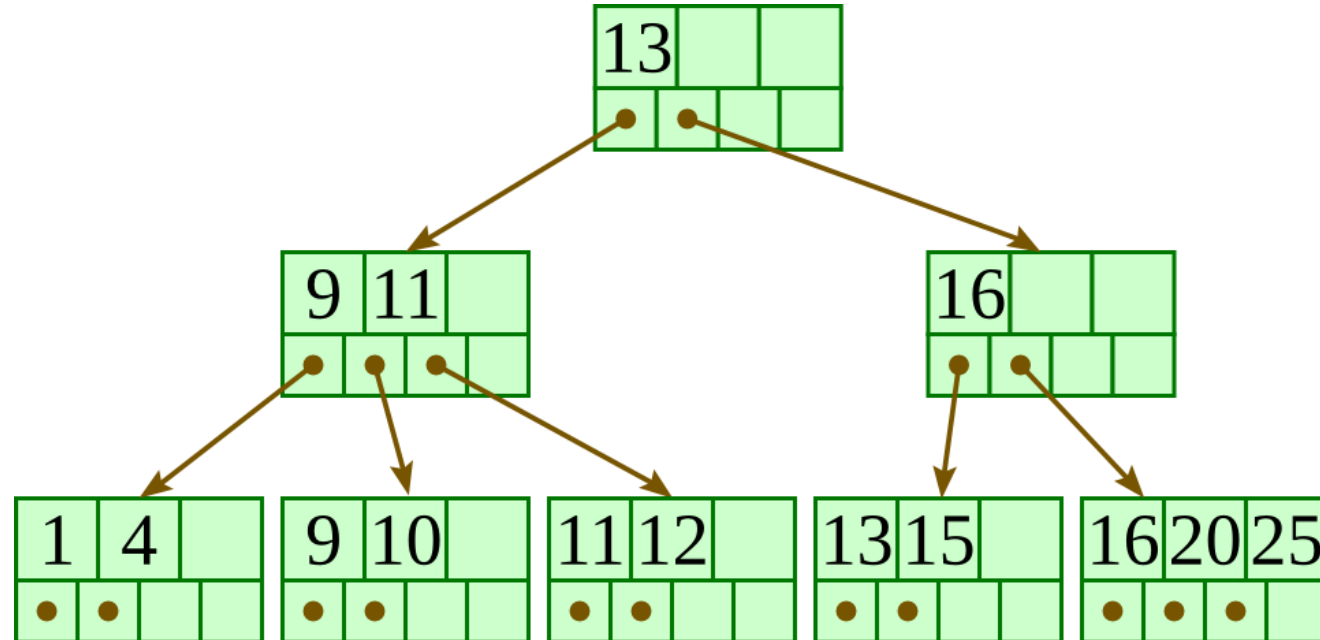
- Минимизируют количество операций чтения-записи с диском;
- Не нужно проходить до конечных узлов, чтобы получить данные;
- Сохраняет ключи в отсортированном порядке для последовательного обхода;
- Использует иерархический индекс для минимизации количества операций чтения с диска;
- Использует частично заполненные блоки для ускорения вставок и удалений;
- Поддерживает сбалансированность индекса с помощью рекурсивного алгоритма.

Недостатки

- Невозможен последовательный доступ;
- Внутренние узлы долго удаляются;
- Поиск не эффективен полностью.

В+-дерево

- В+-tree — структура данных на основе В-дерева, сбалансированное n -дерево поиска с переменным, но зачастую большим количеством потомков в узле.



Преимущества

- Поиск эффективен, так как записи хранятся в конечных узлах;
- Удаление элементов происходит быстро;
- Возможен последовательный доступ к элементам.

Недостатки

- Имеет большую ширину по сравнению с высотой;
- Может быть избыточность в появлении ключей.

B*-дерево

- B*-дерево — разновидность B-дерева, в которой каждый узел дерева заполнен не менее чем на $\frac{2}{3}$ (в отличие от B-дерева, где этот показатель составляет $\frac{1}{2}$).

Преимущества

- Компактнее в сравнении с В-деревом, так как каждый узел используется полнее.

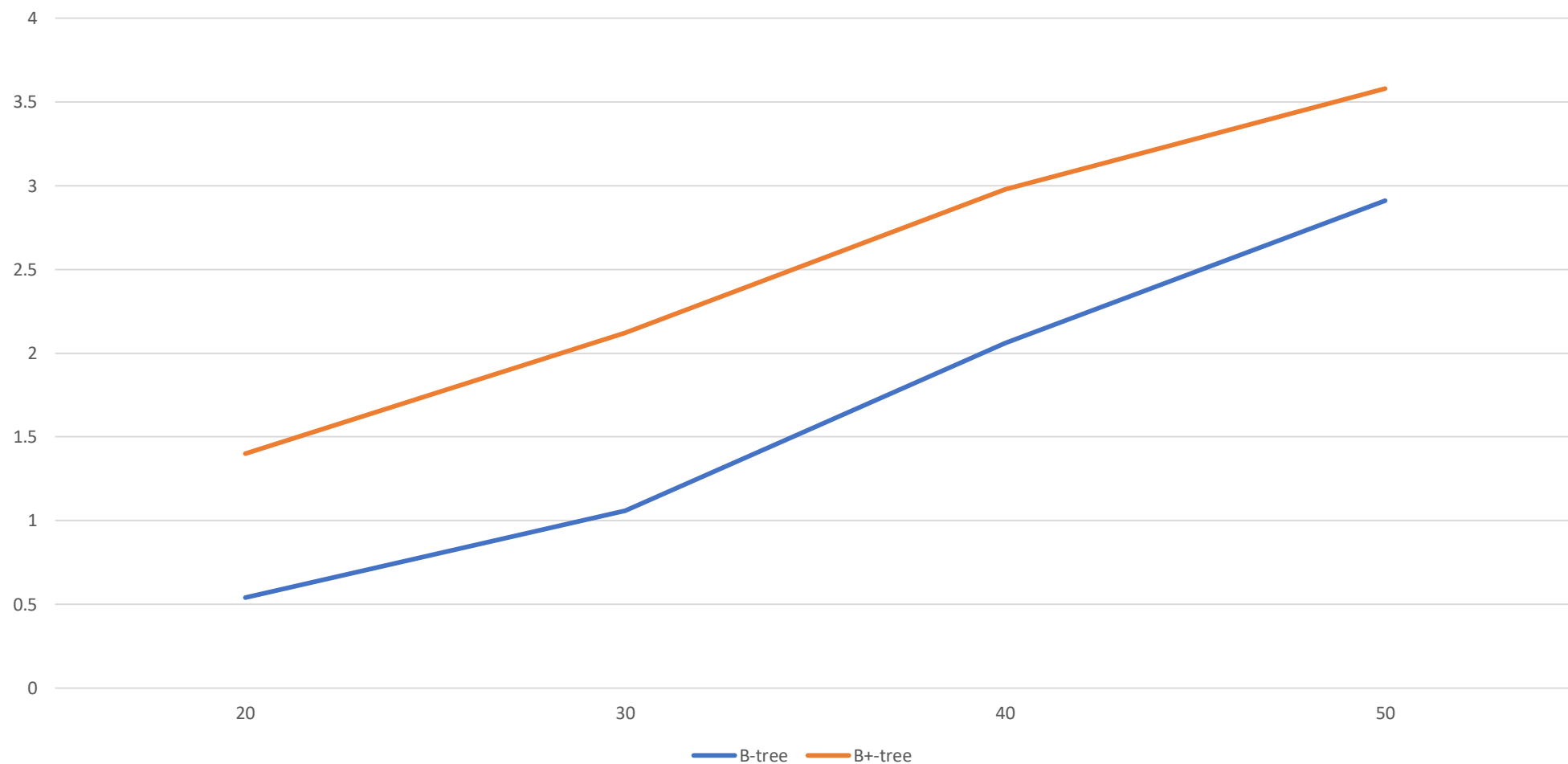
Структура узла

```
struct BTreeNode {  
    int *data;  
    BTreeNode **child_ptr;  
    bool leaf;  
    int n;  
}
```

Гипотеза

Структура данных B-tree быстрее выводит строение дерева на экран, но удобнее в использовании всё же структура данных B+-tree.

Сравнение вывода деревьев на экран в зависимости от количества элементов



Подтверждение гипотезы

Гипотеза подтвердилась: в процессе выполнения практической работы было выявлено множество не особо существенных различий, от которых все же зависит производительность алгоритмов.

Спасибо за внимание!