



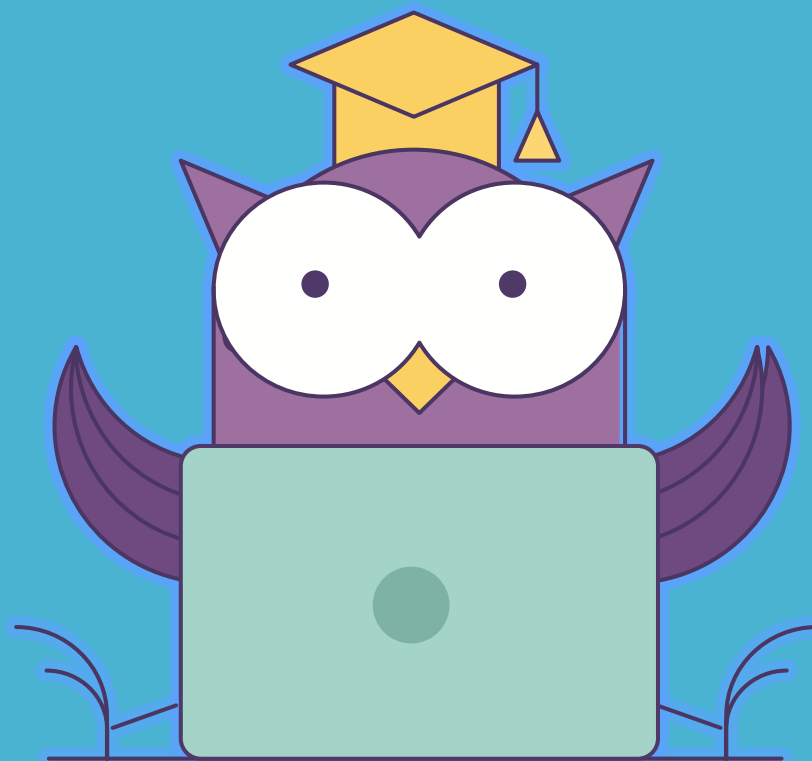
ОНЛАЙН-ОБРАЗОВАНИЕ

Индексы (часть 1)

Юрочко Юрий



Как меня слышно и видно?



> Напишите в чат

+ если все хорошо

– если есть проблемы со звуком или с видео

Юрочко Юрий

- окончил МГТУ им. Н.Э. Баумана в 2016 (ИУ-7)
- 5+ лет разработки на C++/Go
- руководитель команды Go

- Понять, что такое индексы и зачем они нужны.
- Узнать, что такое "бинарное дерево поиска", "сбалансированное дерево".
- Познакомиться с B-деревом и его основными свойствами.
- Научиться создавать и просматривать индексы на примере Postgres.
- Попробовать команду "explain" для анализа запросов.

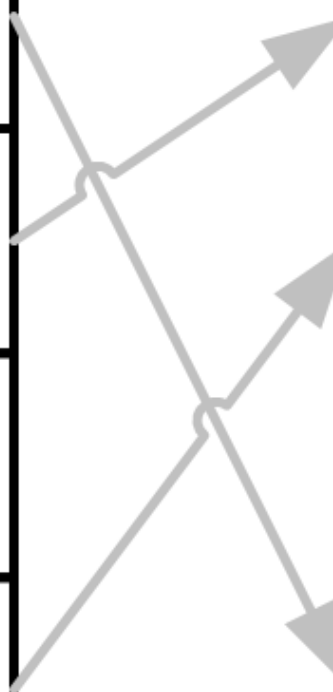
- служебная структура данных
- некий способ отображения ключа в данные
- ускоряют поиск, сортировку
- требуют доп. место для хранения
- обновляются при вставке (модификации данных)

Индекс

Ключ 1	P1
Ключ 2	P2
...	...
Ключ N	PN

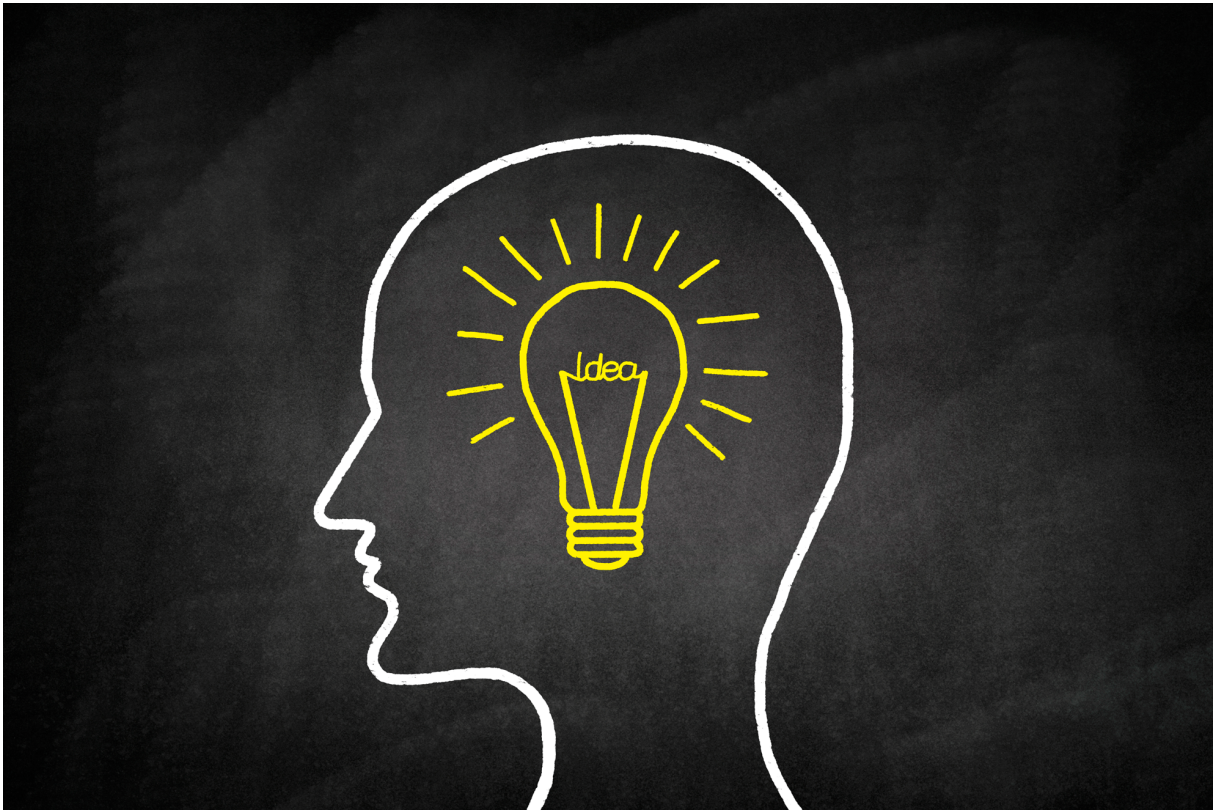
Данные

Запись 1
Запись 2
...
Запись N

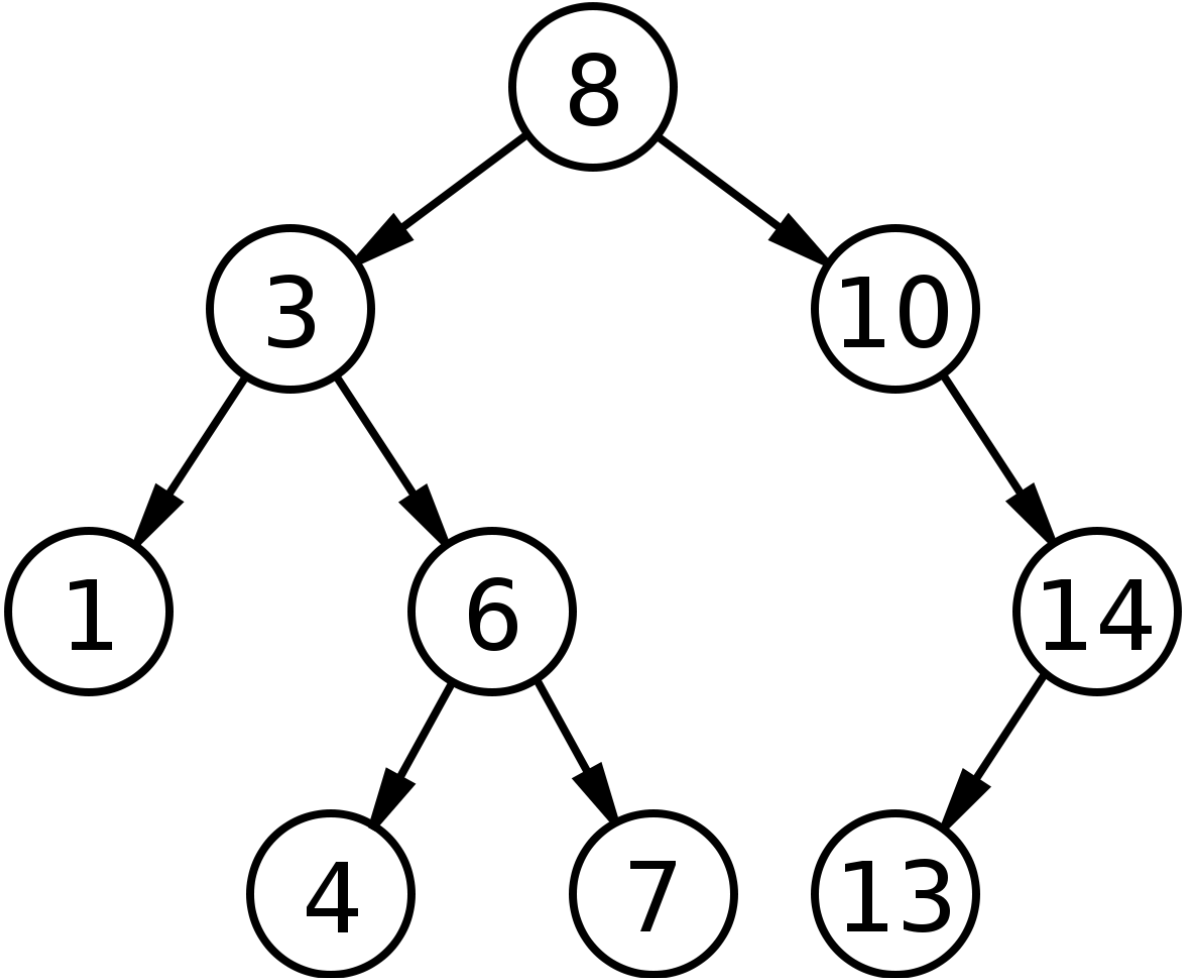


- уникальные и нет
- простые и составные
- кластерные и некластерные

Много индексов - хорошо или не очень?

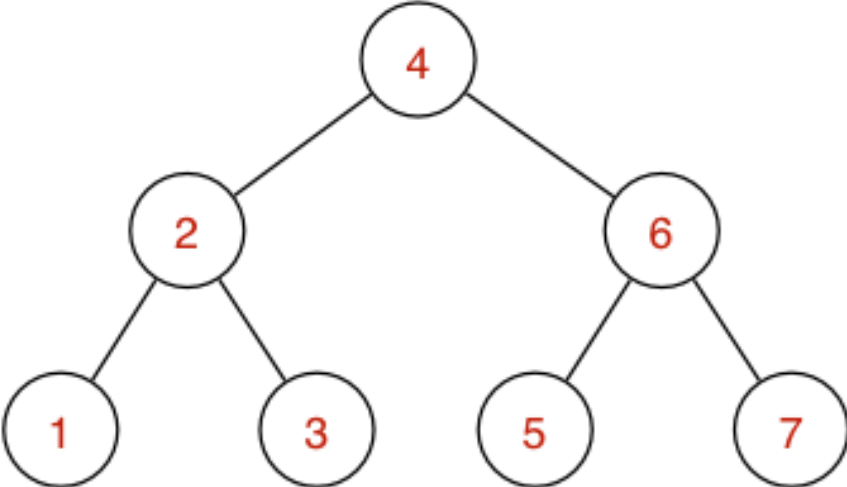




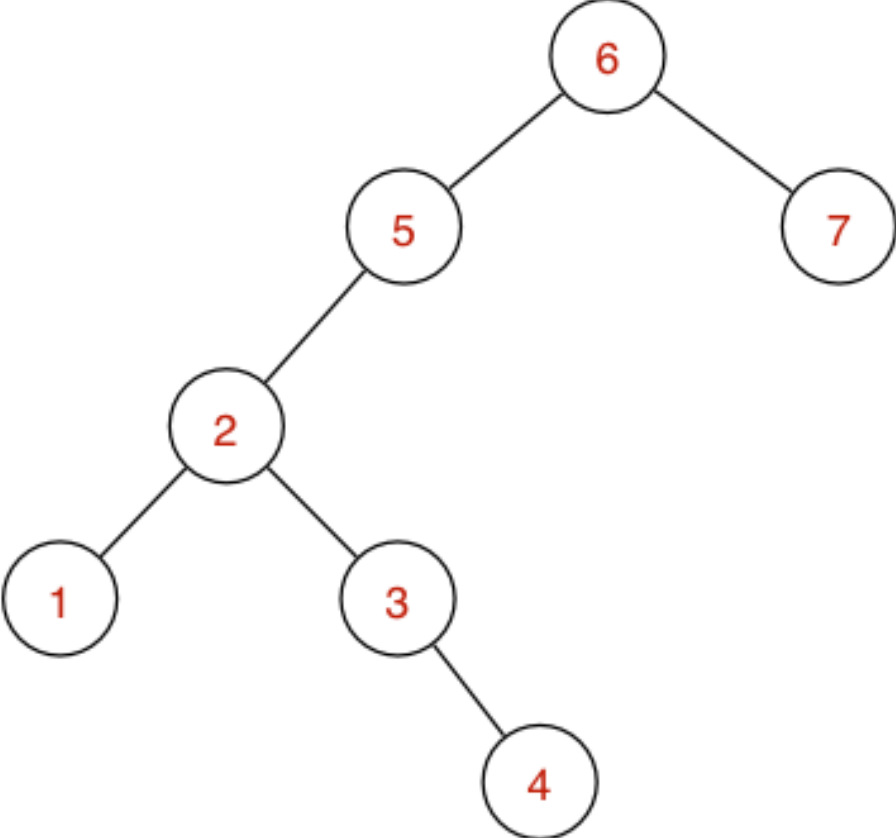


- гарантированный порядок элементов (+)
- понятный поиск (+)
- в вырожденном случае придется посетить все элементы, т.е. $O(n)$ (-)

Balanced Tree



Non-balanced Tree

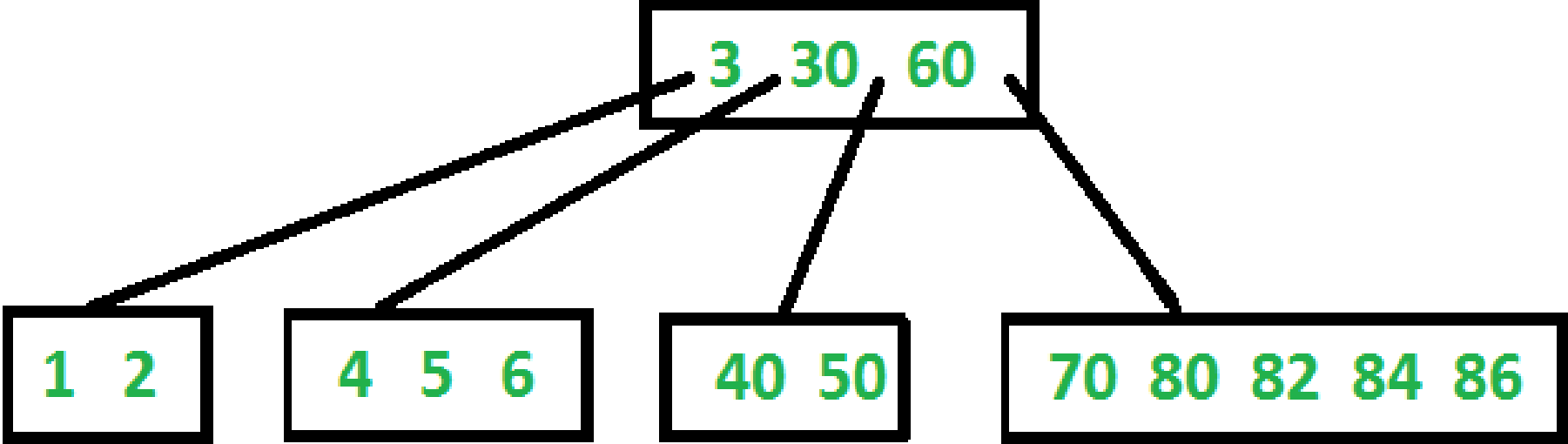


- решает проблему вырожденного случая бинарного дерева (+)
- поиск за $O(\text{высоты дерева})$ (+)
- требует дополнительных усилий на балансировку (-)

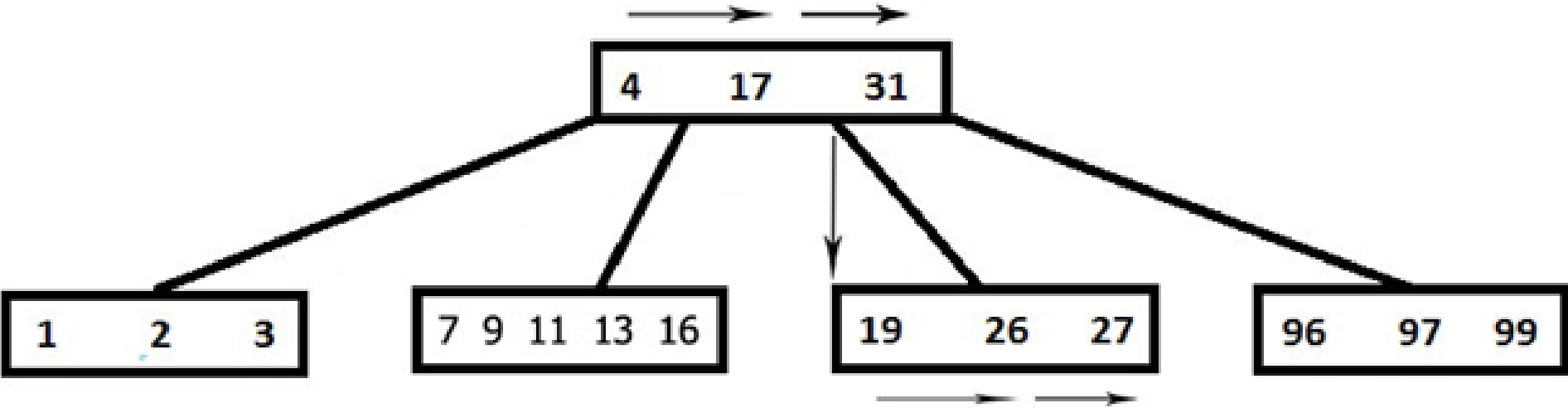
- Вставим 8 - все еще сбалансированное?
- Вставим 9 - все еще сбалансированное?

- оперативная память очень ограничена
- будем использовать внешний носитель (привет, медленный HDD)
- $\log N$ обращений к медленной памяти

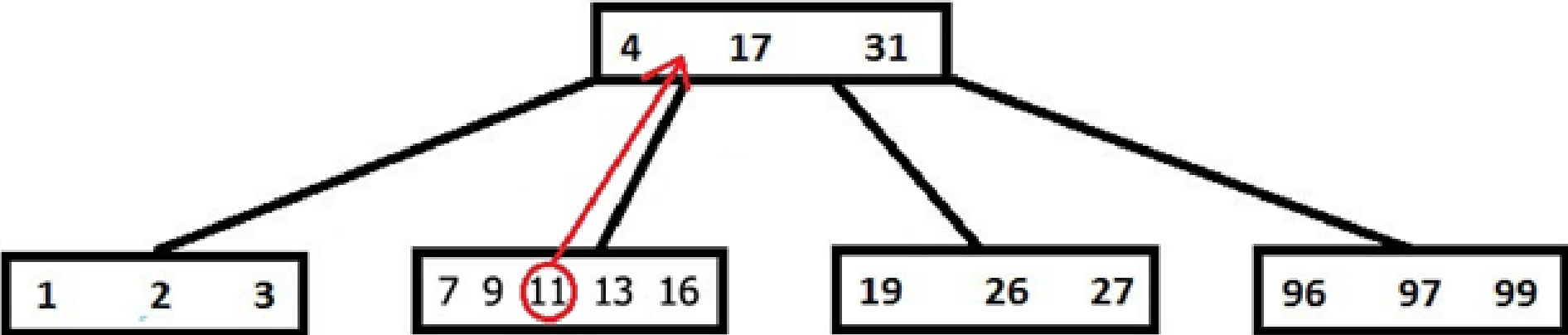
Можем ли мы придумать что-то лучше?!

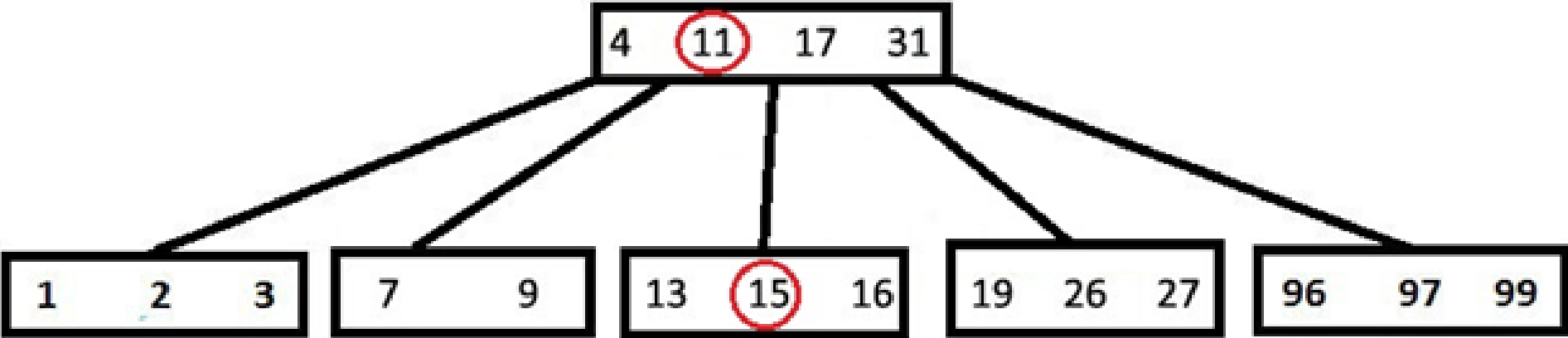


- сильно ветвистое сбалансированное дерево
- t - минимальная степень
- минимум $t - 1$, не более $2t - 1$ ключей (кроме корневого)
- небольшая высота дерева



- алгоритм аналогичен бинарному, но дальнейший выбор не из 2х, а из нескольких
- поиск за $O(t \log t(n))$
- ЧО обращений к диску $O(\log t(n))$





- нельзя добавить ключ в уже заполненный узел
- разбиение на 2 по $t-1$ элементу
- может привести к увеличению высоты
- добавление за $O(t \log t(n))$
- ЧО обращений к диску $O(\log t(n))$

- больше $t \Rightarrow$ меньше высота дерева
- зависит от размера блока на диске
- зависит от объема оперативной памяти
- обычно от 50 до 2000
- $t=1001$ и 1млрд записей \Rightarrow 3 операции для любого ключа

Внимание на экран

- <https://www.youtube.com/watch?v=KFcpDTpoixo&feature=youtu.be>
- <https://habr.com/ru/company/postgrespro/blog/326096/>
- <https://www.geeksforgeeks.org/introduction-of-b-tree/>

- Поняли, что такое индексы и для чего они нужны.
- Узнали, что такое "бинарное дерево поиска", "сбалансированное дерево".
- Познакомились с B-деревом и его основными свойствами.
- Создали и просмотрели индексы на примере Postgres.
- Попробовали команду "explain" для анализа запросов.

Вопросы?

<https://otus.ru/polls/5524/>



Спасибо за внимание!

