

В-деревья

Определение

- В-дерево изобретено в 1972г. Р.Байером и Е. Маккрайтом
- Предназначено для создания мелких деревьев для быстрого доступа к диску
- Мелкие деревья имеют мало уровней; продвинуться к цели по такому дереву можно, выполнив небольшое количество проходов.
- В-деревья — это мощное решение проблемы дискового хранения; фактически каждая коммерческая система баз данных уже давно использует вариации В-деревьев

Определение

- В-дерево состоит из *страниц*.
- Каждая страница имеет набор индексов.
- Каждый индекс содержит значение ключа и указатель.
- Указатель в индексе может указывать либо на другую страницу, либо на данные, сохраняемые в дереве.

Определение

- Страница называется *узловой*, если индекс страницы указывает на другую страницу
- Страница называется *листовой* если индекс указывает на данные, (от слова "листва").
- Максимальное количество индексов на странице называется *порядком* страницы

Определение

- Каждая страница максимально может иметь количество дочерних страниц, равное ее порядку.

Для B-деревьев существует правило:

- ни одна из страниц, кроме верхней и листовых, не может иметь индексов, количество которых меньше половины порядка (**order/2**).
- листовая страница может иметь меньшее количество **ИНДЕКСОВ (order/2 - 1)**

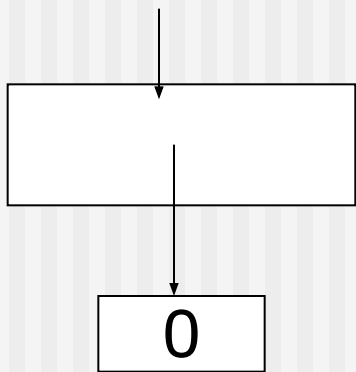
Определение

- Новые индексы всегда добавляются в листовые страницы.
- ***Вы никогда не добавляете индекс к узловой странице.***
- Узловые страницы создаются В-деревом при разбиении существующих.

Построение B-дерева: 1.

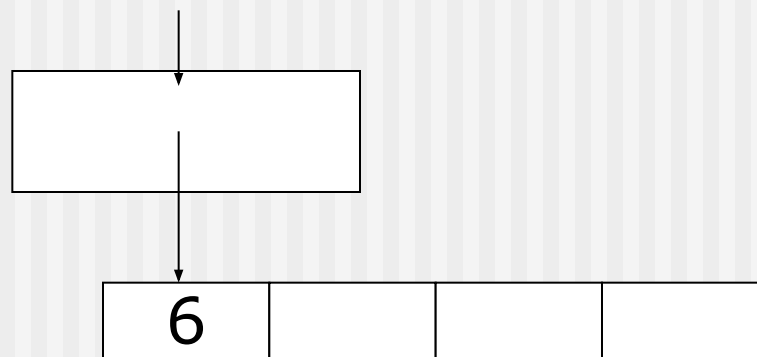
- 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень



Пустое
дерево

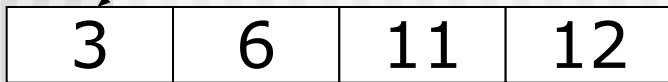
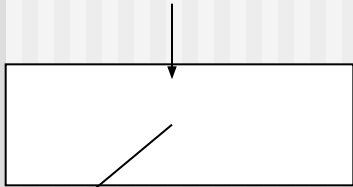
Корень



Построение B-дерева: 2-3.

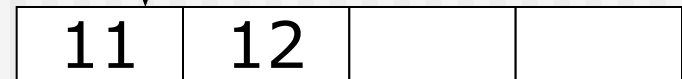
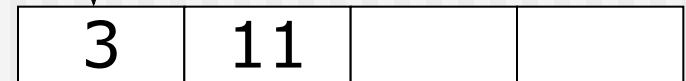
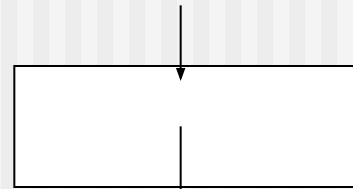
- 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень



Заполненная
листовая страница

Корень



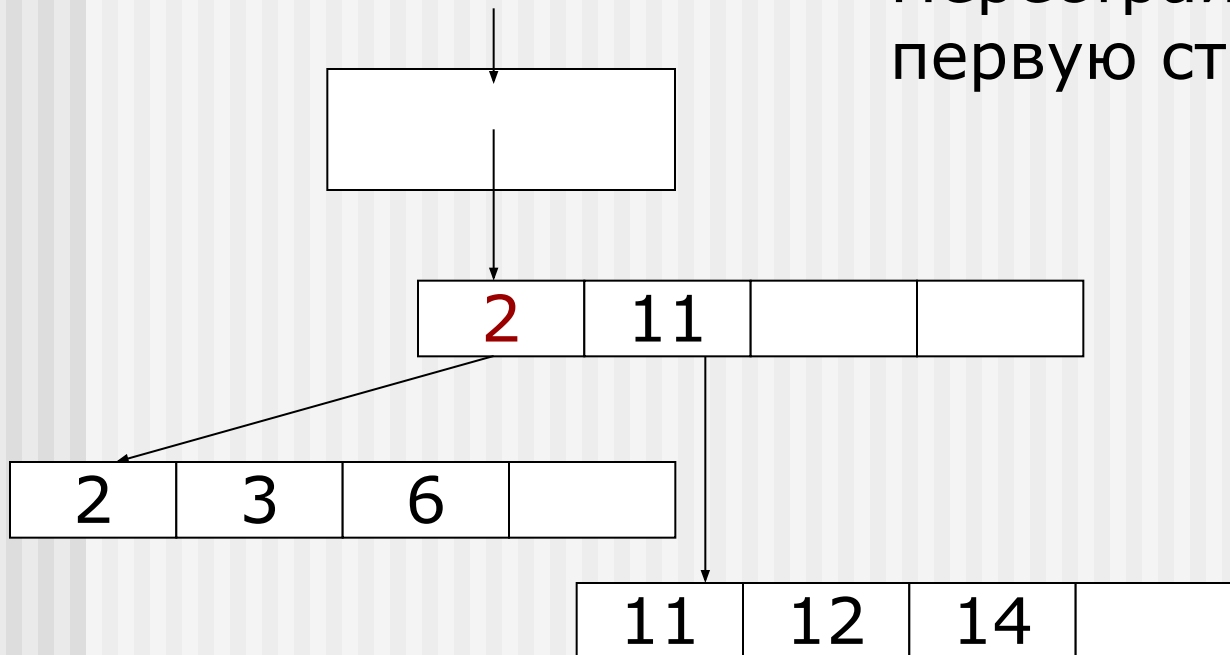
Разбиваем страницу на две

Построение В-дерева:

- 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень

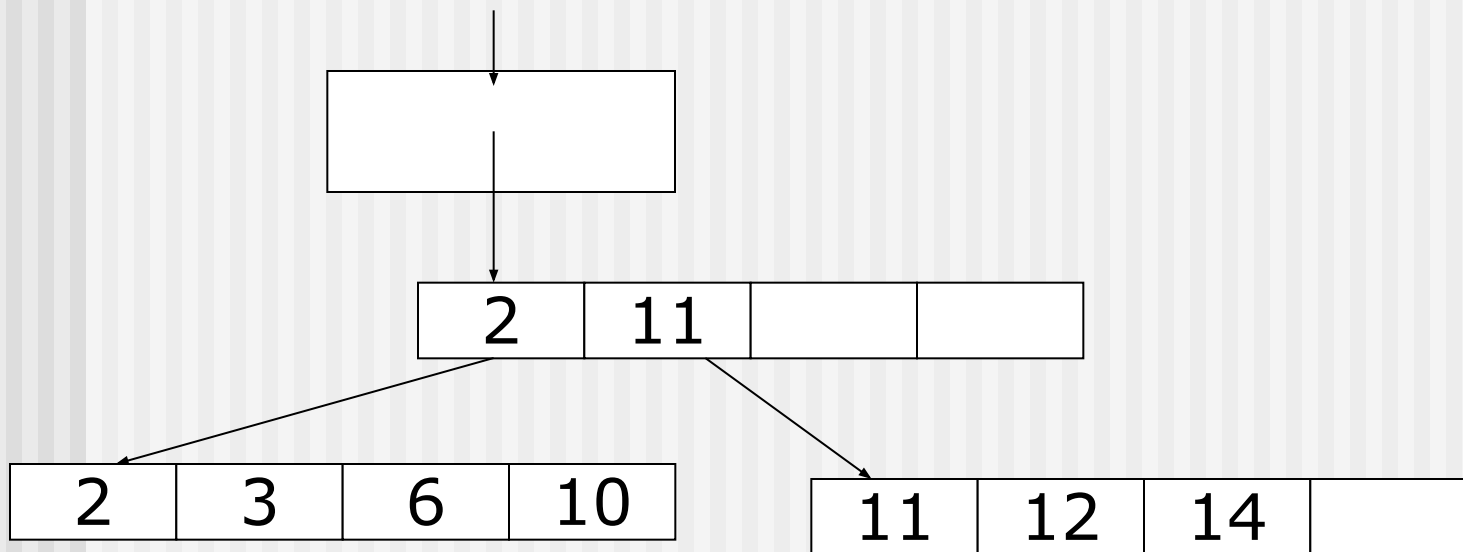
Перестраиваем
первую страницу



Построение B-дерева:

- 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень

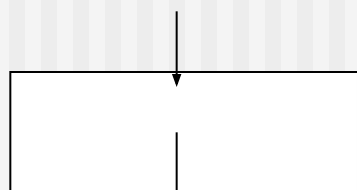


Разбиваем страницу

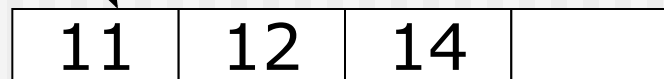
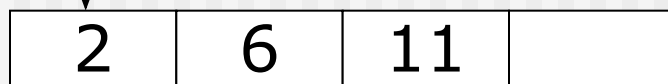
Построение В-дерева:

- 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень



К корневому указателю добавляем новый и перестраиваем страницу

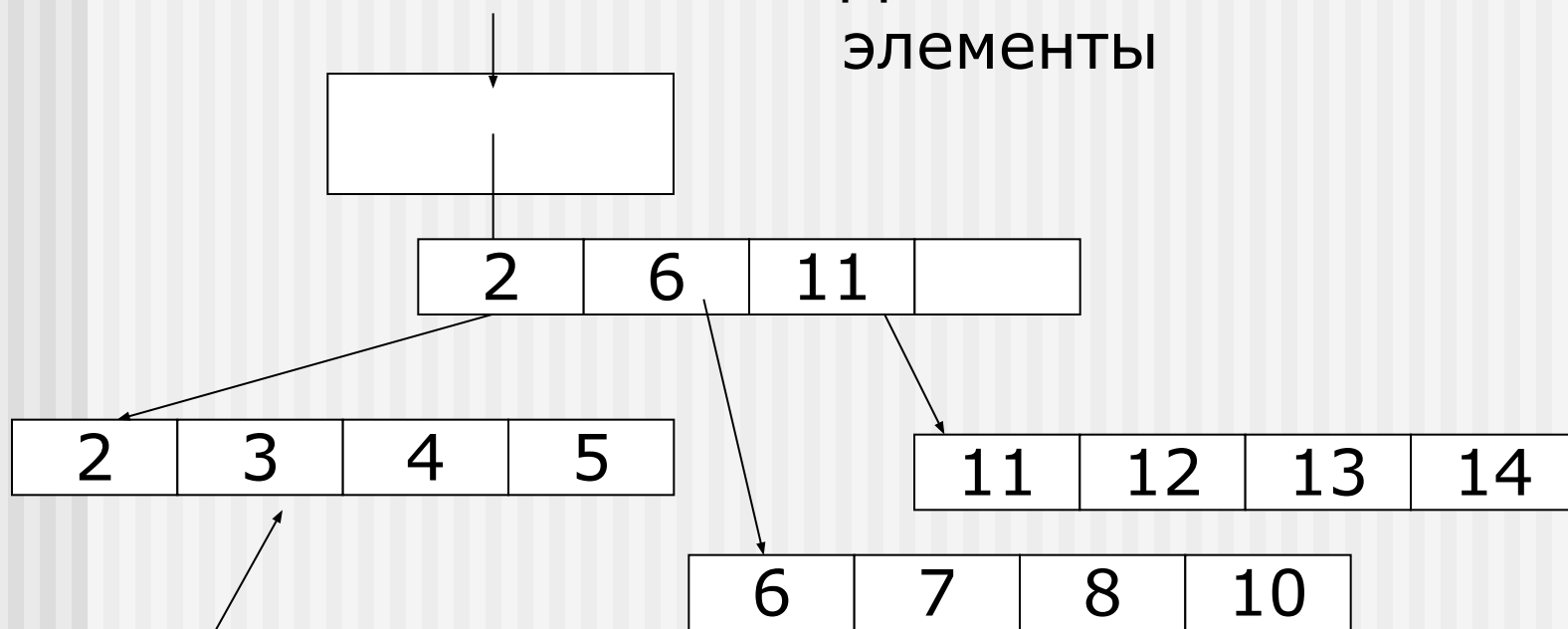


Построение B-дерева:

- 6 11 3 12 14 2 10 5 4 7 8 13 19

Корень

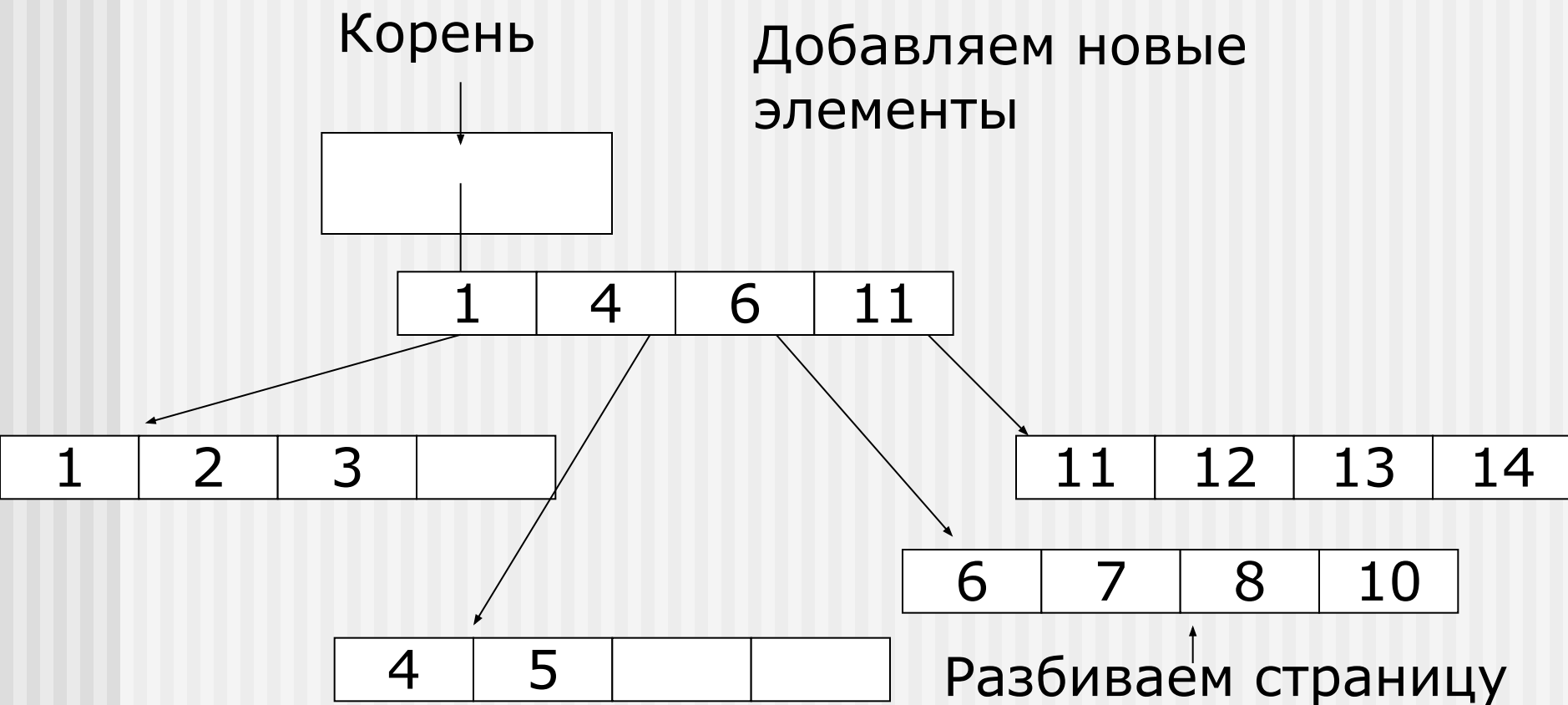
Добавляем новые
элементы



Разбиваем заполненные страницы и добавляем указатель к верхней

Построение B-дерева:

- 6 11 3 12 14 2 10 5 4 7 8 13 19



Построение B-дерева:

■ 6 11 3 12 14 2 10 5 4 7 8 13 1 9

Корень

Разбиваем верхнюю
страницу



Алгоритм добавления новой страницы

1. Разбить страницу пополам.
2. Добавить новый ключ в подходящую позицию
3. Установить соответственно указатель.
4. Вернуть указатель на новую страницу.
5. Если корень определит, что требуется новая верхняя (узловая) страница, создать ее.
6. В новую верхнюю страницу добавить запись, на которую будет указывать корень.
7. В новой верхней странице добавить запись для возвращаемого значения шага 4.
8. Указать корню на новую верхнюю страницу.

Запись B-дерева на диск

- Общее предназначение B-дерева состоит в сохранении данных на диске
- B-дерево должно сохранить на диске страницы, индексы и данные

Размер страниц

- Цель: быстрое считывание с диска
- Большинство ПК работает быстрее, если считываются блоки размером кратным 2
- Идеальный размер определяется размером сектора жесткого диска

Размер страниц

- Будем рассматривать размер – 512
- Каждая запись индекса должна быть делителем порядка, чтобы на каждой странице помещалось четное число
- Длина индекса будет составлять 16 байтов: 4 байта на указатель (теперь смещение) и 11 байтов на данные, с завершающим строку байтом NULL.

Размер страниц

- На странице размером 512 байтов существует 32 набора по 16 байтов (т.е. 32 объекта индекса).
- Порядок B-дерева составляет 32.
- 32-порядковое дерево может содержать 1024 слова в двух уровнях, 32768 слов в трех уровнях и 33554432 слов в пяти уровнях.
- В большинстве случаев поиск может завершиться за несколько обращений к жесткому диску, что идеально..

Количество страниц, которые одновременно могут находиться в памяти

- для определения числа страниц, которые одновременно могут находиться в памяти необходимо сделать обход начинающаяся с верхнего узла и прокладывая себе путь вниз по дереву.
- Поскольку каждая страница может содержать 32 индекса, то в любое время половина из них будет задействована: 16 индексов на страницу.
- 10 уровней страниц с 16 индексами на страницу предоставляют триллион ключей.

Количество страниц, которые одновременно могут находиться в памяти

- 10 страниц, впрочем, занимают всего лишь 2 Кб памяти. (16 байтов на индекс умножить на 16 байтов на страницу и умножить на 10 страниц, равно 2560 байтов, или 2 Кб.)
- Вот и мощь В-деревьев: постоянно занимая в памяти всего 2 Кб, они обеспечивают доступ к триллиону ключей!