

# (9.) В-деревья

Имеет более <sup>двух</sup> ~~одного~~ потомков, но отличается от других n-ных деревьев.

его  $\oplus$  в том, что оно (или) получается "небесными" и хорошо подходит для хранения на диске.

Сбалансированное дерево поиска, которое <sup>и</sup> ~~может~~ каждый узел содержит множество ключей и имеет более двух потомков.

Кол-во ключей зависит от порядка В-дерева

В-дерево m-порядка имеет след. свойства

1.) Глубина всех листьев одинакова.

2.) Все узлы, кроме корня должны иметь  $\frac{m}{2} - 1$  ключей и максимум  $m-1$  ключей

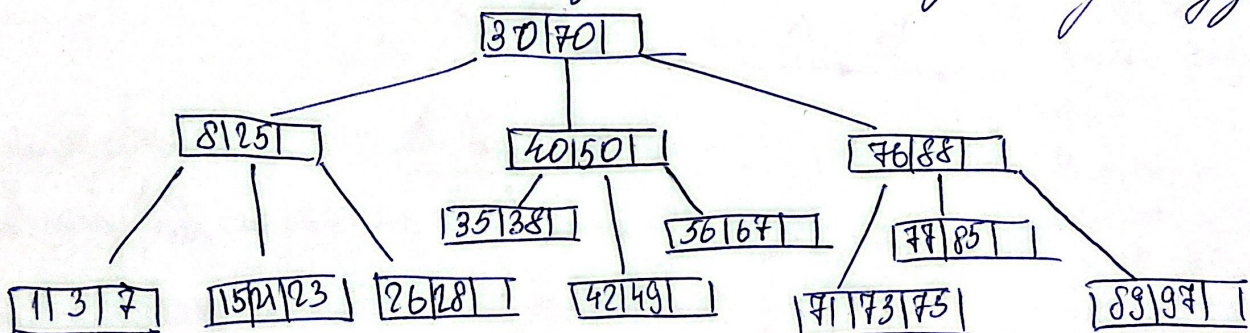
3.) Все узлы без листьев кроме корня, должны иметь минимум  $\frac{m}{2}$  потомков.

4.) Если корень - это узел не содержащий листьев, он должен иметь 2-потомка

5.) Узел без листьев с n-1 ключами должен иметь n потомков.

6.) Все <sup>ключи</sup> ~~узлы~~ в ~~каждом~~ узле должны располагаться в порядке возрастания их значений.

Пример: В-дерево 4-го порядка имеет 3 значения ключа и максимум 4 потомка для каждого узла.





Смывая в том, чтобы мы меньше обращались к массиву  
длин, т.к. он медленный.

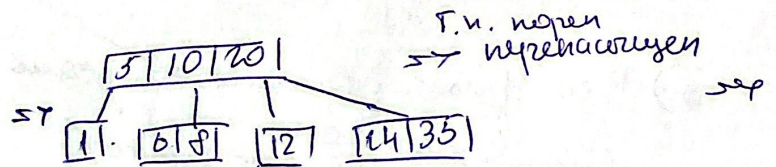
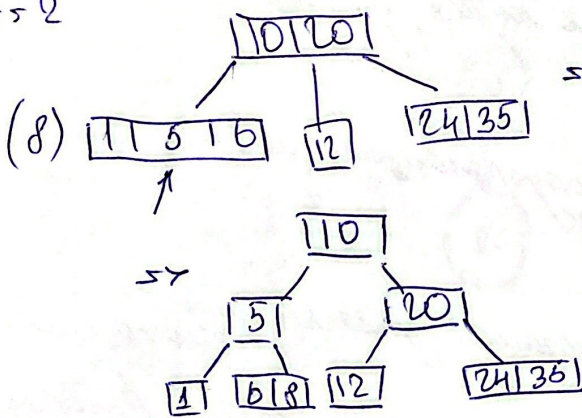
## Поиск $O(\log n)$

- 1) Сравниваем элемент первой узле-массиве с нашим искомым элементом, например бинар. поиском.
- 2) Далее также сравниваем в "степе"  $C_i$  узле-массиве, и так пока не найдём элемент.

## Вставка

- 1.) Поиск  $\rightarrow$  2.) Проверка на возможность вставки элемента в  
мест  $\rightarrow$  3.) Если можно, то вставляем на  
свое место;  $\rightarrow$
- 4.) Если элементов в листе равно  $2^k - 1$ , тогда берём центральный  
и вставляем его в "отца":

$k=2$



т.к. перед  
 $\rightarrow$  переносим