Multiway Search Trees

Outline

- Basic concept
- Properties and operations
- B-trees
- Key insertion
- Key removal

Basic Concept

Number of Nodes vs. Number of Levels

- Given 3 nodes in a BST
  - Tallest: 3 levels
  - Shortest: 2 levels
- Given 7 nodes in a BST
  - Tallest: 7 levels
  - Shortest: 3 levels
- Given 1000 nodes in a BST
  - Tallest: 1000 levels
  - Shortest: 10 levels

Simple Arithmetic

- Observation from BST
  - $2^{Level} - 1$
    - $2^{20} - 1$: 1M nodes
- More nodes
  - $3^{Level} - 1$
    - $3^{20} - 1$: 3486784400 nodes
    - 3^{13}: 1 M nodes
  - $4^{Level} - 1$
    - $4^{20} - 1$: 1T nodes
    - 4^{10}: 1M nodes

Issue of Storage

- How to pack more data items?
- More items per level
  - More subtrees
- More items per node

Efficiency in Tree Storage

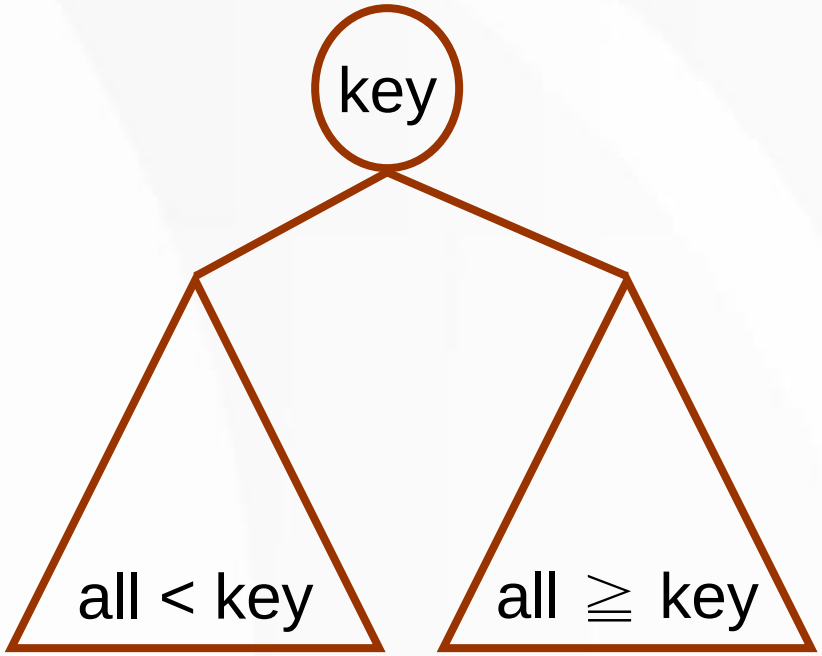
- Efficient storage
  - Balanced BST
  - AVL Trees
- Issue
  - Height of a tree
  - Shorter the better
- Goal
  - Complete
  - Nearly complete tree

Beyond BST

- Observation
  - maximum number of data items per node
  - maximum number of nodes per level
- BST
  - one data item per node
  - two subtrees under a node
- Multiway Search Tree
  - k data items per node
  - k+1 subtrees under a node

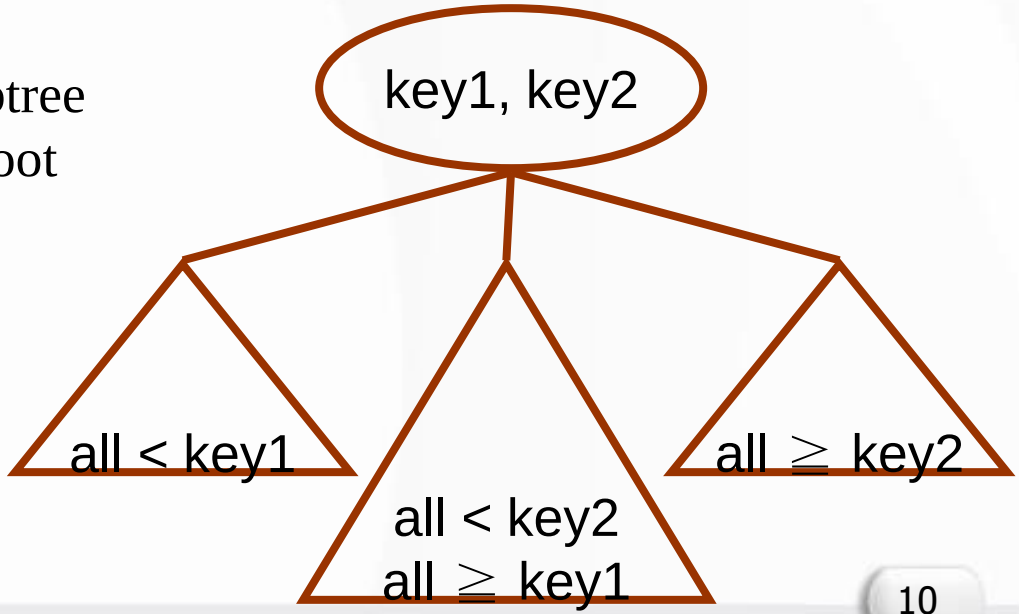
Definition of BST

- Binary tree
- Relationship among values in nodes
  - All values in the left subtree < value in the root
  - All values in the right subtree ≧value in the root
- For all subtrees

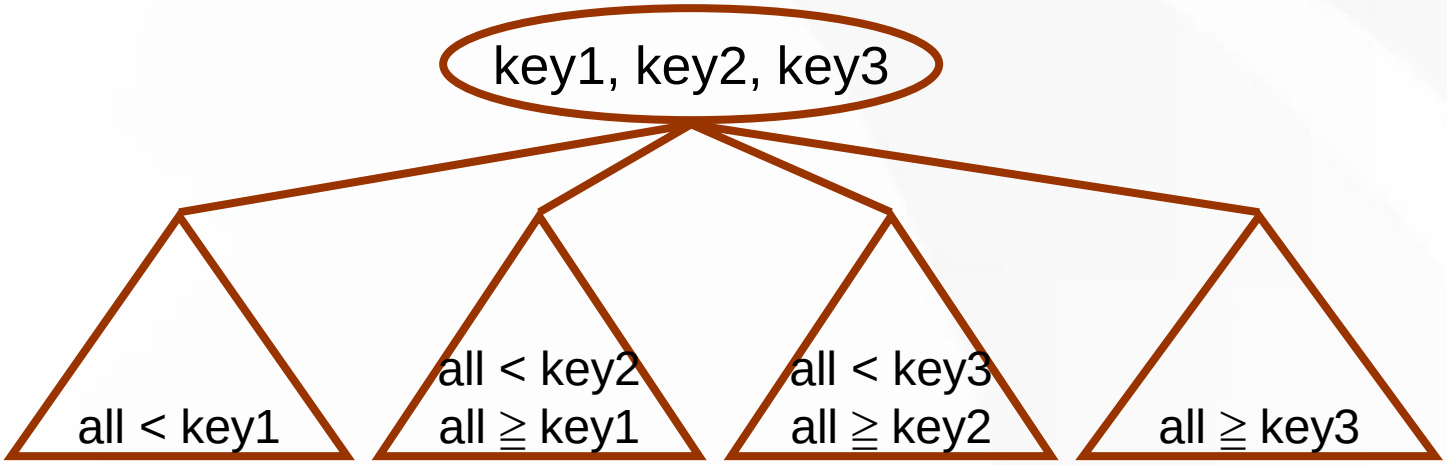


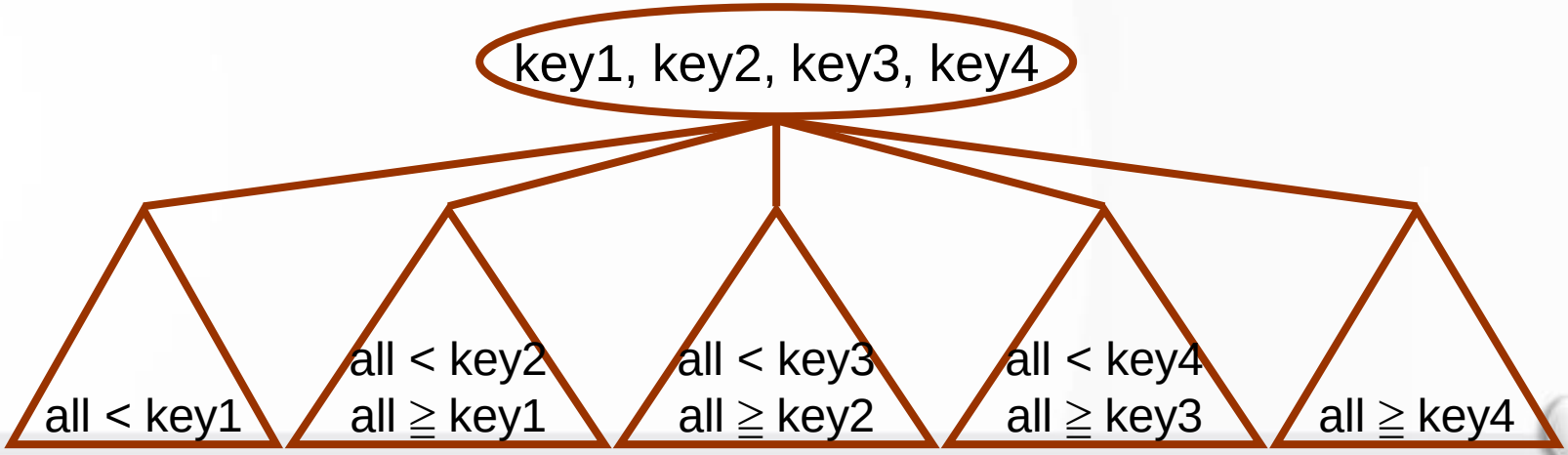
Extension to the Definition of BST

- Ternary tree??
- Relationship among values in nodes
  - All values in the left most subtree < key1 in the root
  - All values in the middle subtree ≧ key1 and < key2 in the root
  - All values in the right most subtree ≧ key2 in the root
- For all subtrees

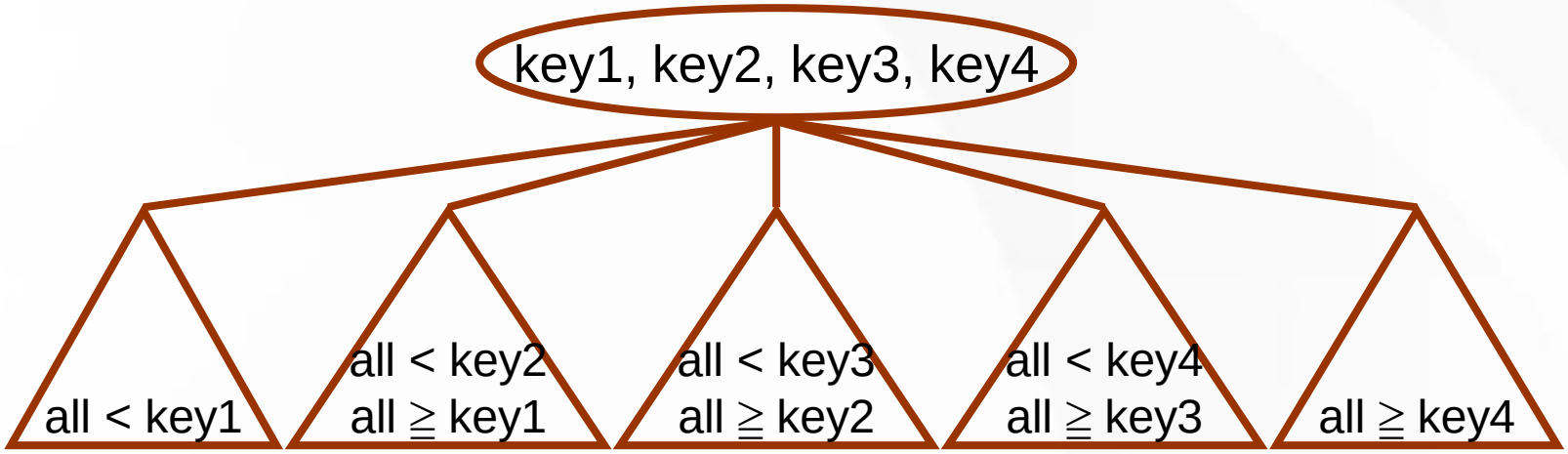


Extension to the Definition of BST





Observation



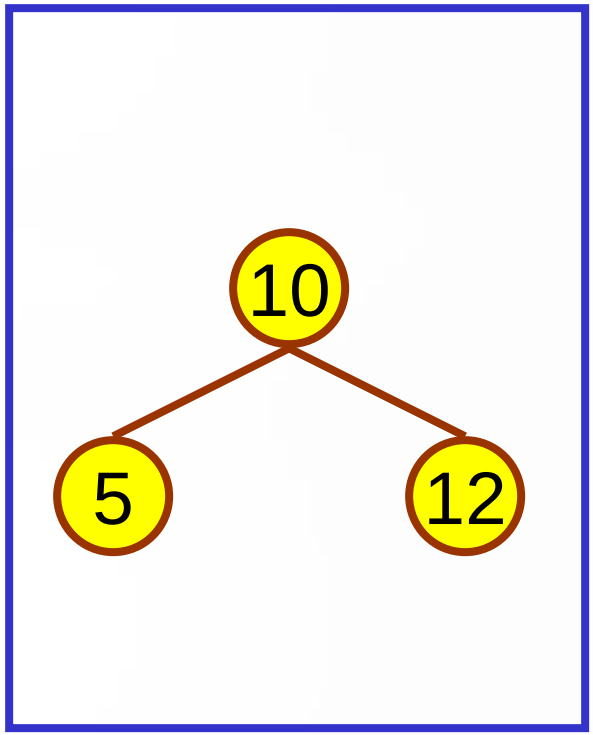
- key1 < key2 < key3 < key4
- 4 keys
  - 5 subtrees

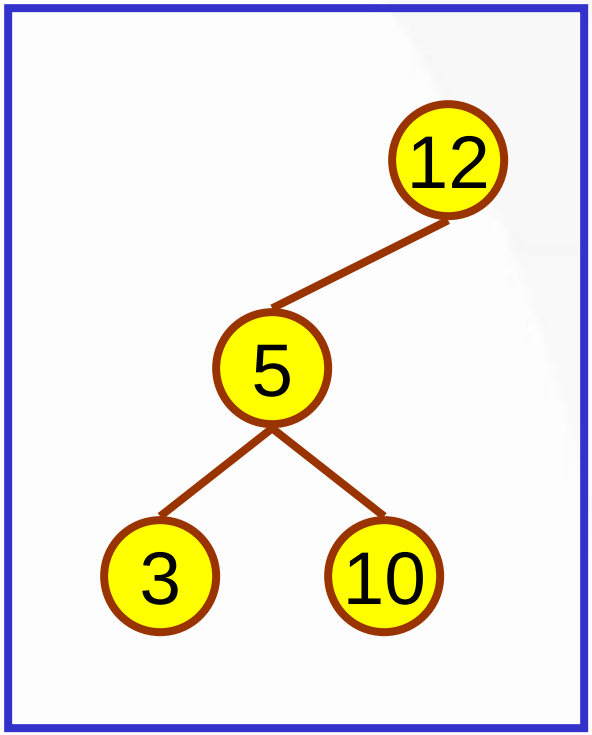
Concept of Order

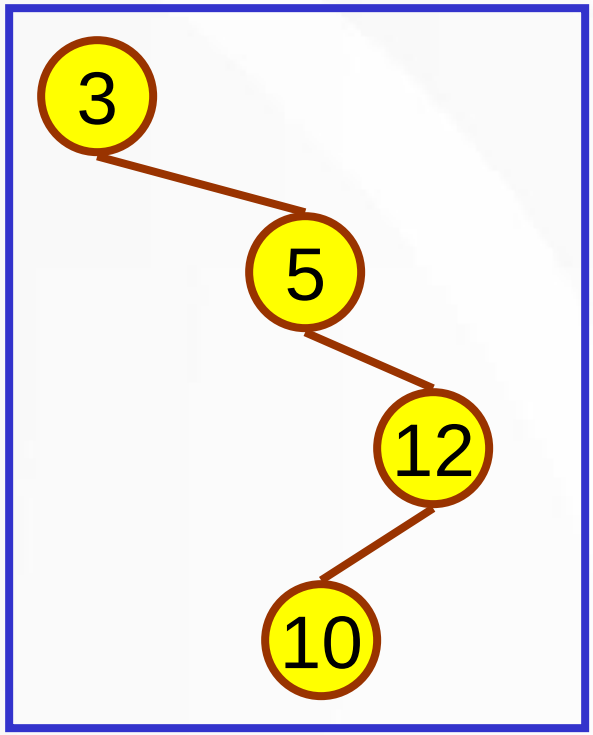
- N keys
  - N+1 subtrees
- Order (or degree)
  - Number of subtrees
- BST
  - Order of 2
- M-way tree
  - Order of m

Order	# of Subtrees	# of Keys
1	1	?
2	2	1
3	3	2
…	…	…
m	m	m – 1

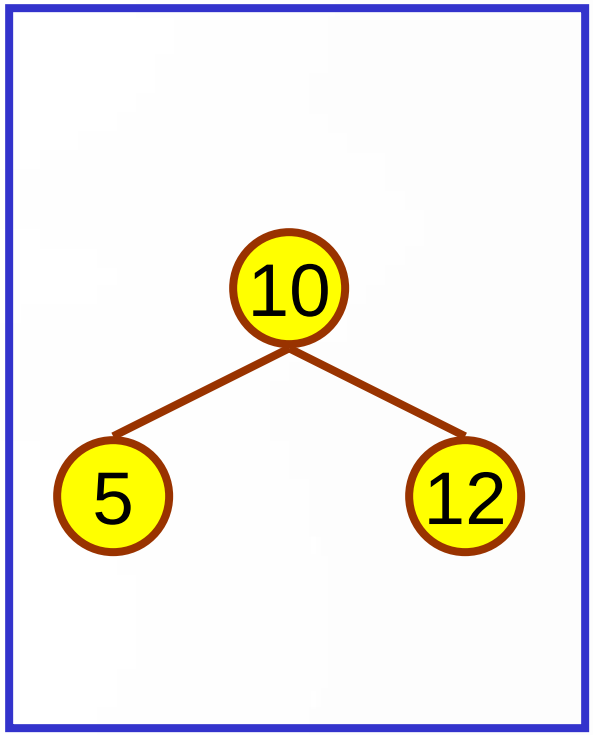
Examples of a BST



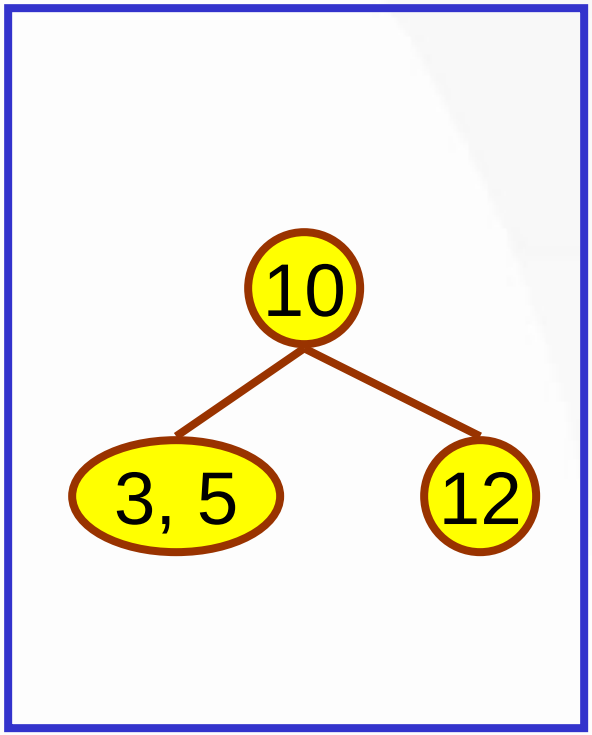




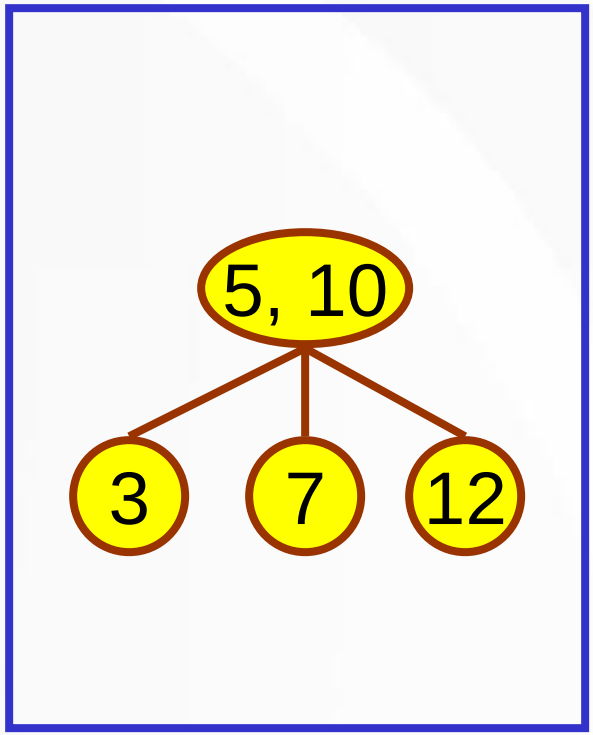
Examples of a Multiway Search Tree



Order = 2



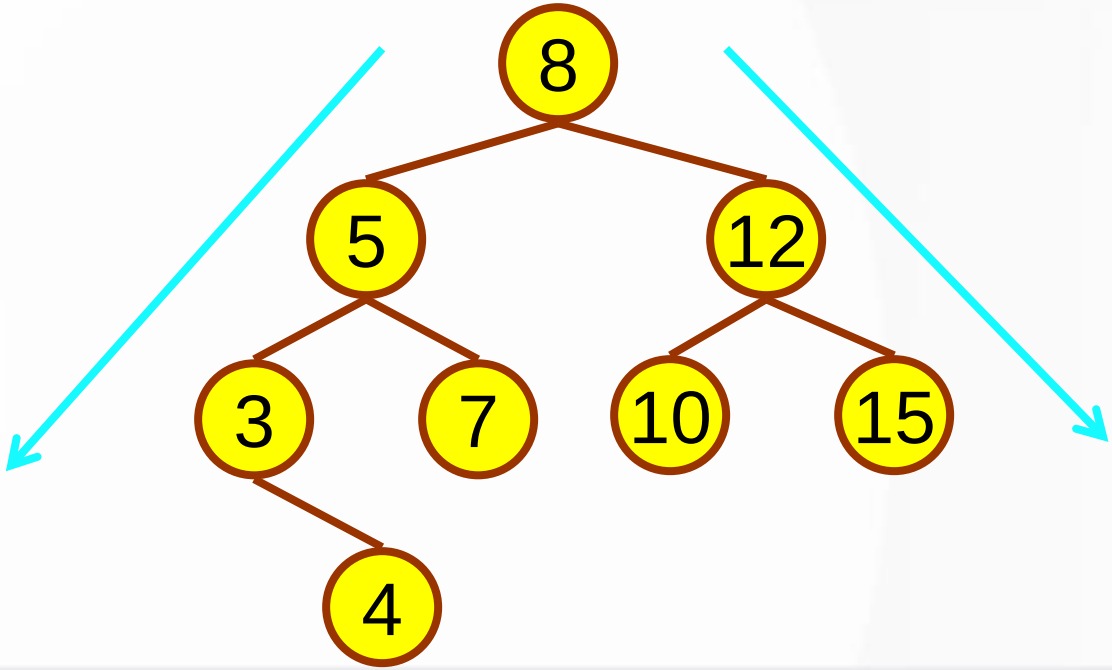
Order = 2



Order = 3

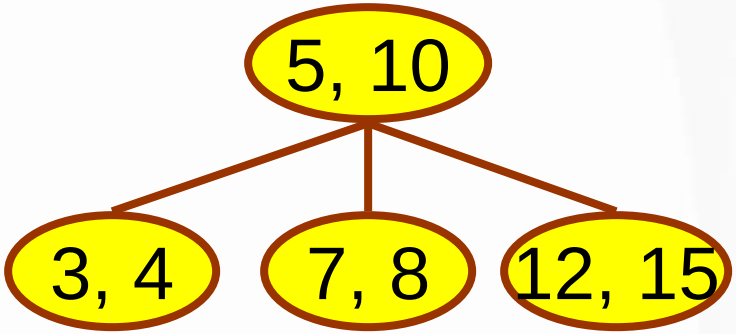
Observation in a BST

- Where is the smallest node?
- Where is the largest node?



Observation in a Multiway Search Tree

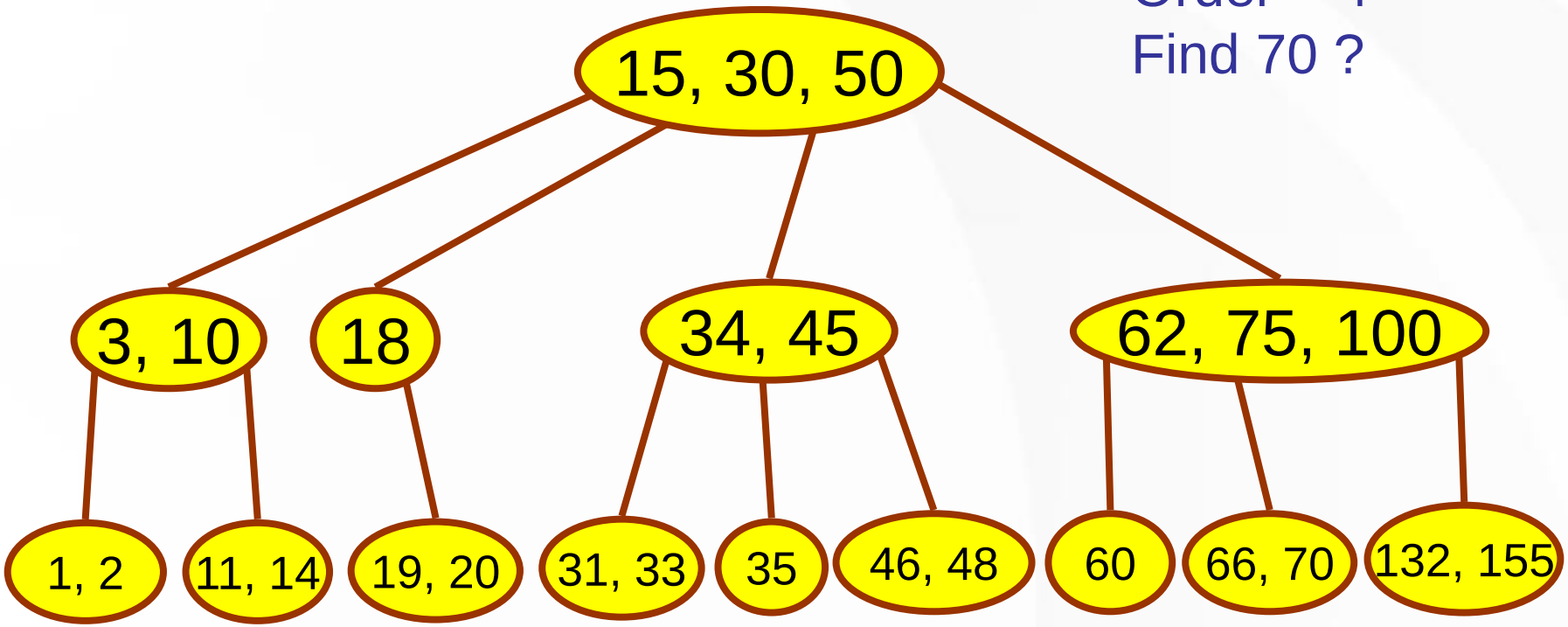
- Where is the smallest node?
- Where is the largest node?



Properties and Operations

Example

Order = 4
Find 70 ?

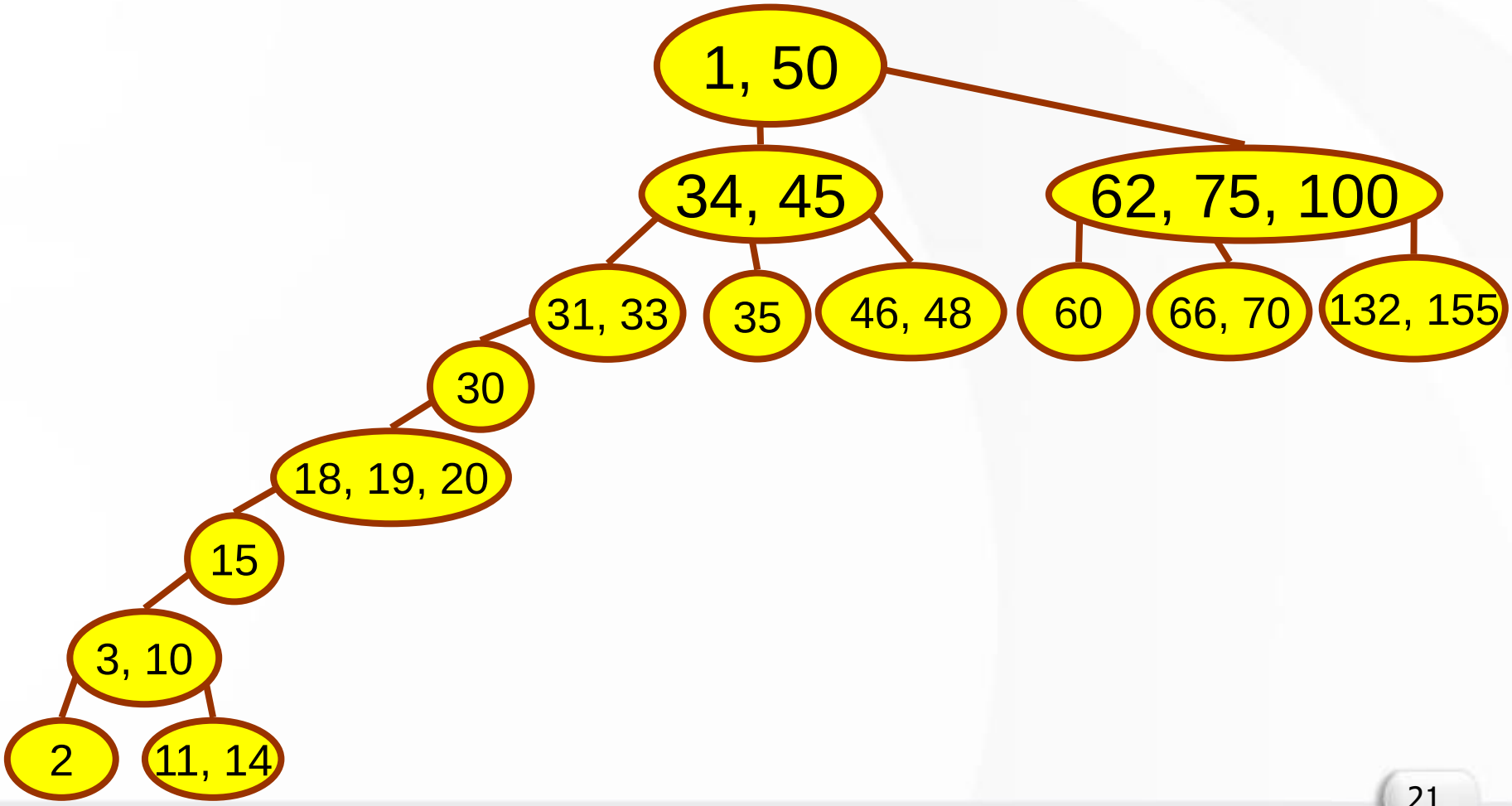


Search Operation

- Start from the root
- Search in the root
- If not found
  - Go to a subtree
  - Search in the root
  - ...

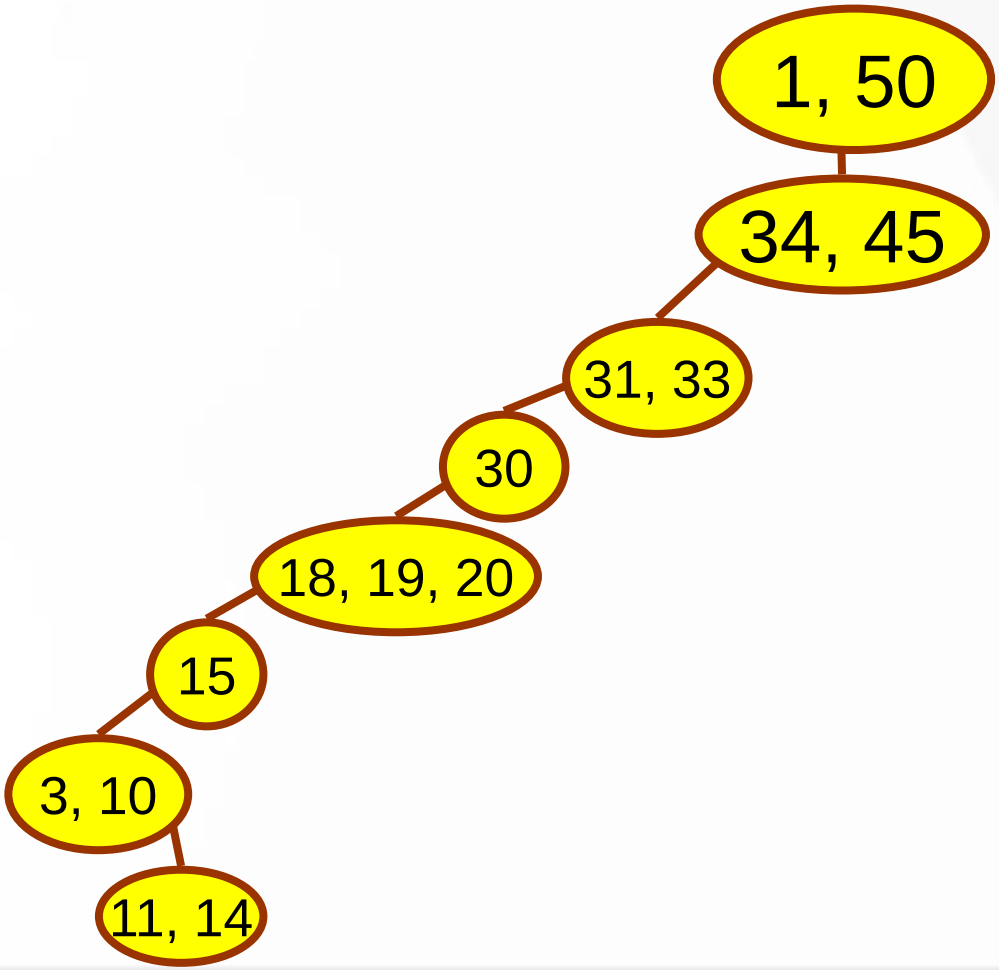
```
Search_mTree (mTree T, data key)
{
    if T is empty
        return FALSE

    found ← seqSearch (T's Root, key)
    if found
        return TRUE
    else {
        S← findSubT (T's Root, key)
        Search_mTree (S, key)
    }
}
```

Example (more extreme version)



B-Tree

Tilted Tree



Complete or Nearly Complete Tree

- Efficiency in search
  - Quickly finding (or not finding) the target
- Efficiency in storage
  - Fully packing each level
  - Most number of nodes

Review: Constructing a Tree

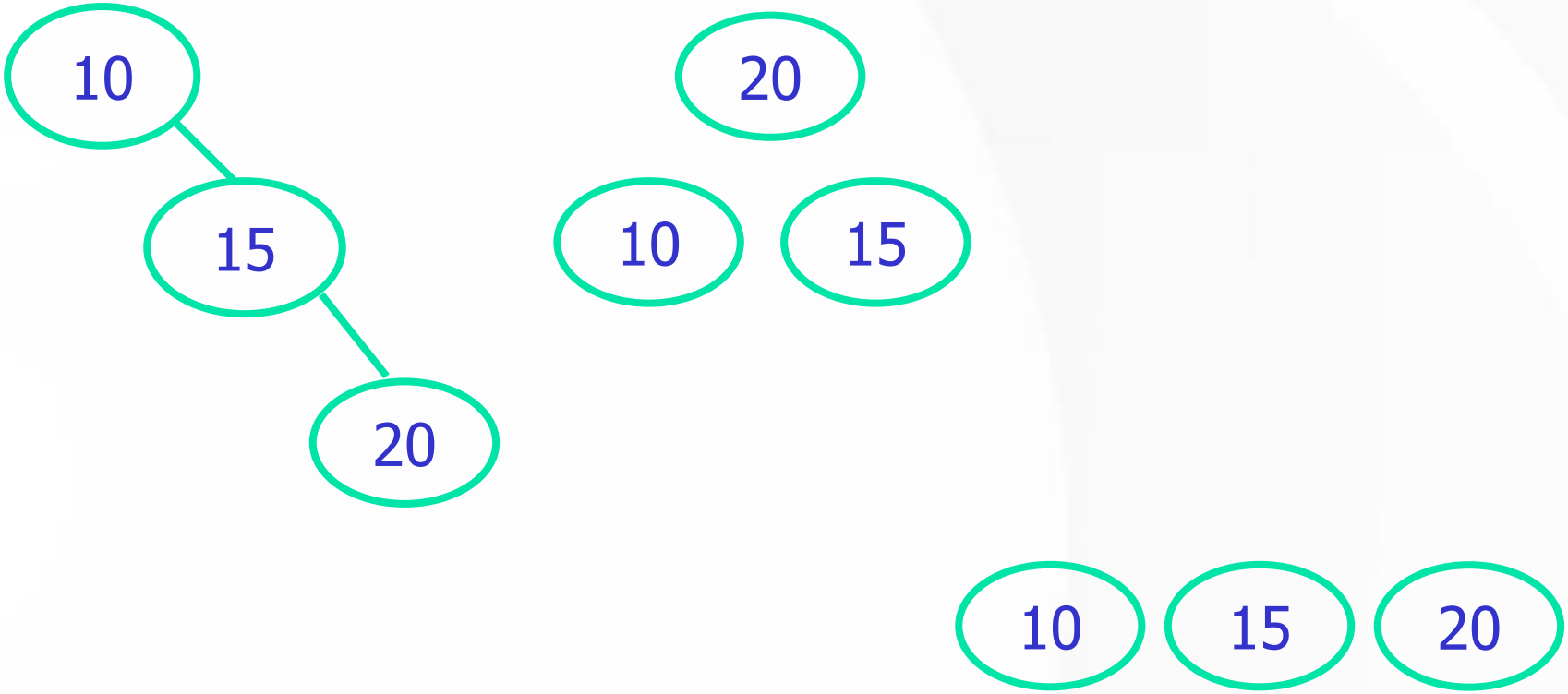
- Original idea
  - Top down
  - Root → Internal nodes → Leaf nodes
  - “Insert” takes place at bottom.

Different Ways to Construct a Tree

- Original idea
  - Top down
  - Root → Internal nodes → Leaf nodes
  - “Insert” takes place at bottom.
- New idea
  - Bottom up
  - Leaf nodes → Internal nodes → Root
    - Filling up the base
    - Stacking level by level

Illustration

Key sequence: 10, 15, 20



Observation on Nodes

- Keys per node
  - Empty node
  - Full node
  - In-between node
- Nodes per level (number of subtrees)
  - Empty node
  - Full node (k keys)
    - k+1 subtrees
  - In-between node
    - 1 key: 2 subtrees
    - 2 keys: 3 subtrees
    - k-1 keys: k substrees

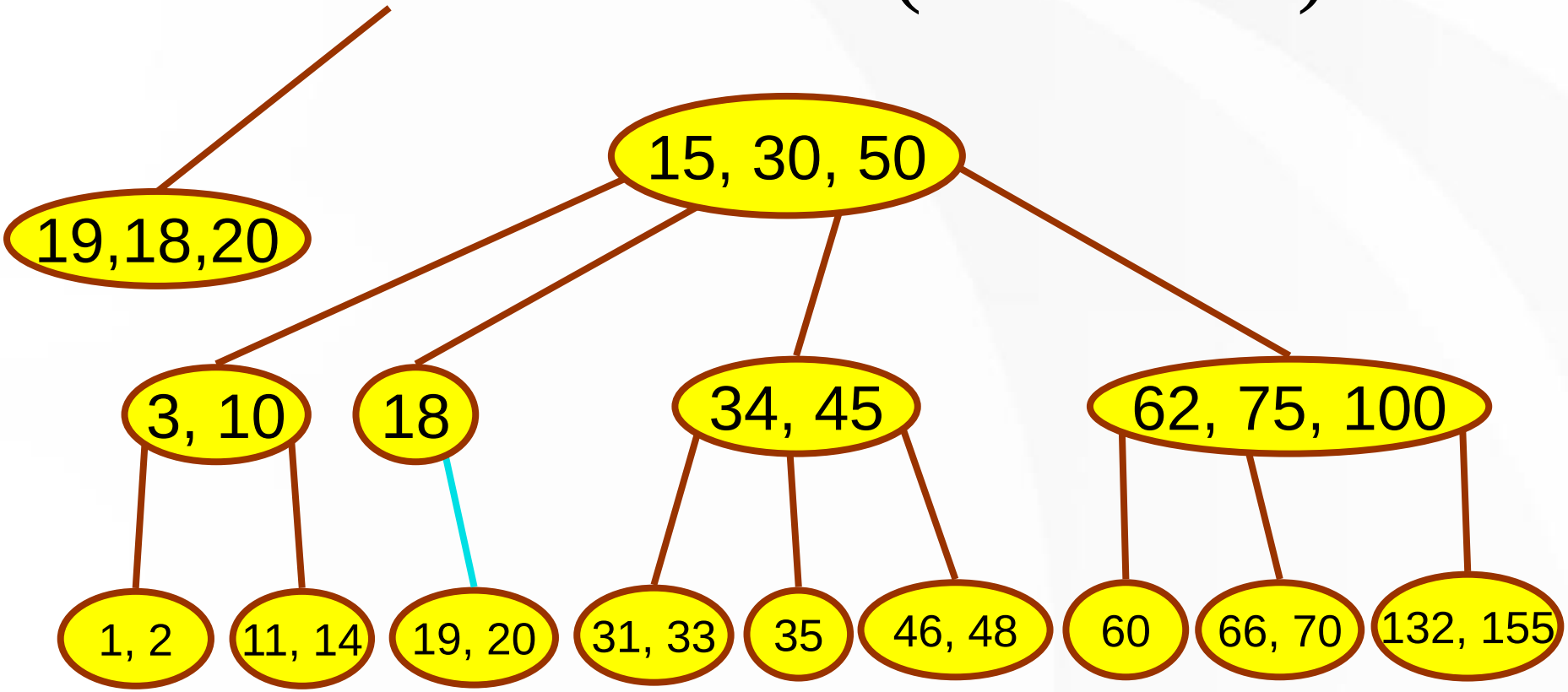
Order of B-Trees

Order	Number of Subtrees	
	Minimum	Maximum
3	2	3
4	2	4
5	3	5
6	3	6
…	…	…
m	$\lceil (m/2) \rceil$	m

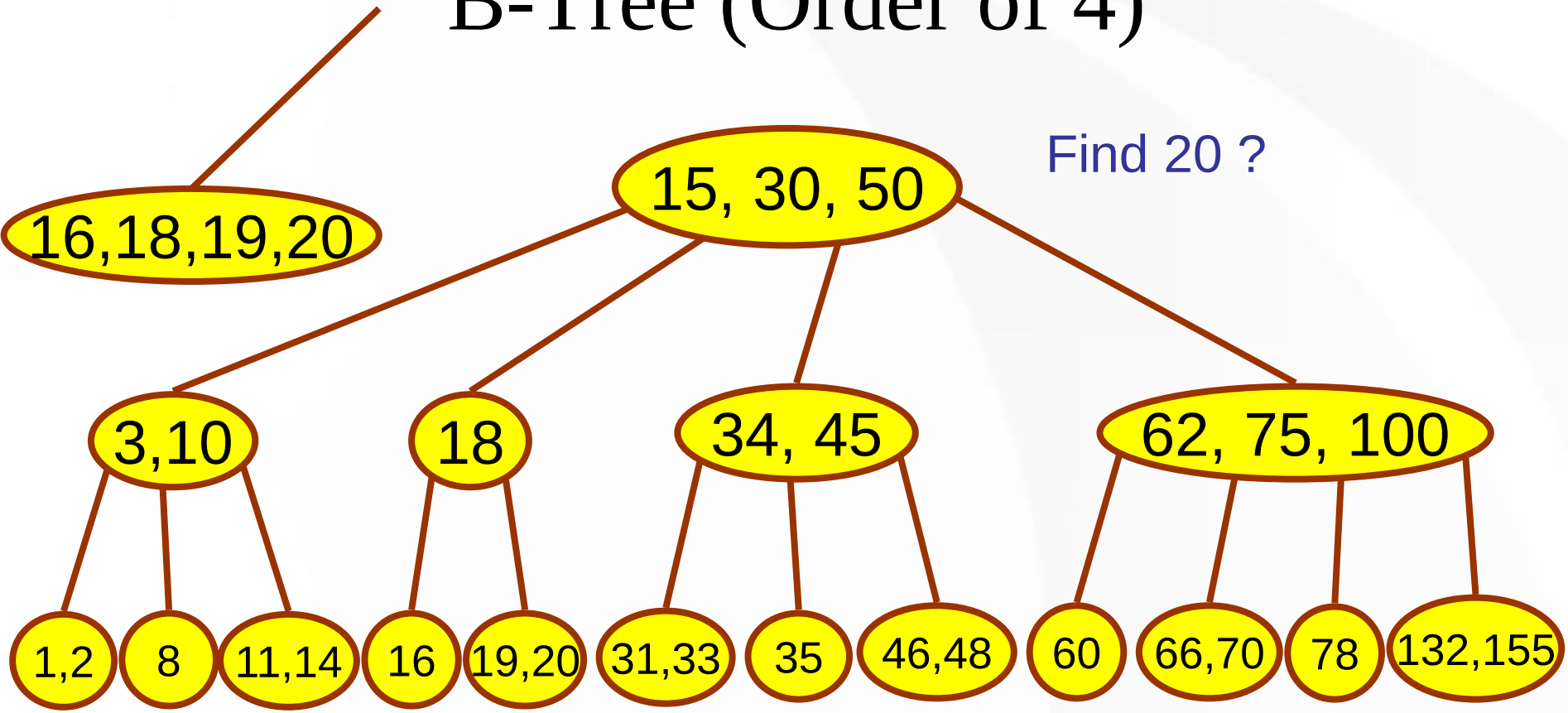
Order of B-Trees (number of keys)

Order	Number of Subtrees		Number of Keys	
	Minimum	Maximum	Min	Max
3	2	3	1	2
4	2	4	1	3
5	3	5	2	4
6	3	6	2	5
…	…	…	…	…
m	$\lceil (m/2) \rceil$	m	$\lceil (m/2) \rceil - 1$	m-1

Incorrect B-Tree (Order of 4)



B-Tree (Order of 4)



Key Insertion

Insertion

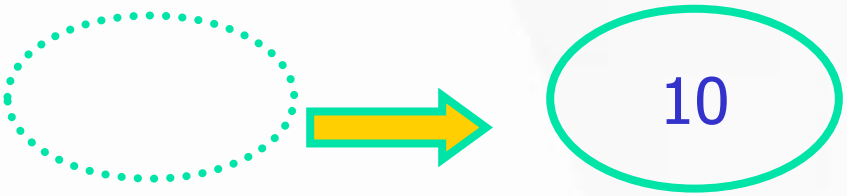
- Where to insert?
  - BST
    - new node as a leaf or a leaf-like node
  - Multiway tree
    - **new node or within an existing node** at the **leaf** level
- Cases to consider
  - Empty tree
  - Tree with nodes
- Empty tree
  - New node → Root

Insertion

- Tree with nodes
  - Find the correct **leaf node**
  - Insert a data item in a node
    - Node not full
      - Insert to an ordered list
    - Node full
      - Insert to a sequential ordered list (overflow)
      - Promote the key in the middle
      - Split the node

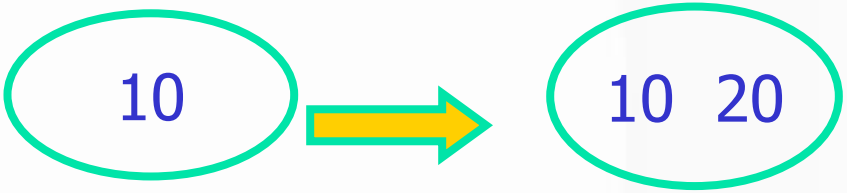
Illustration – Order of 3

- To insert 10, 20, 15
- Empty node



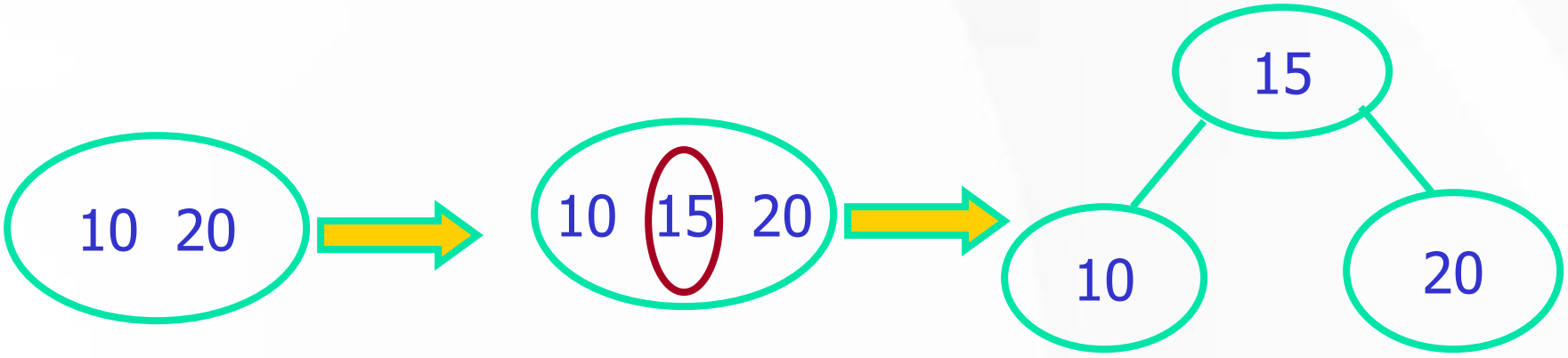
to insert 10

- Node not full

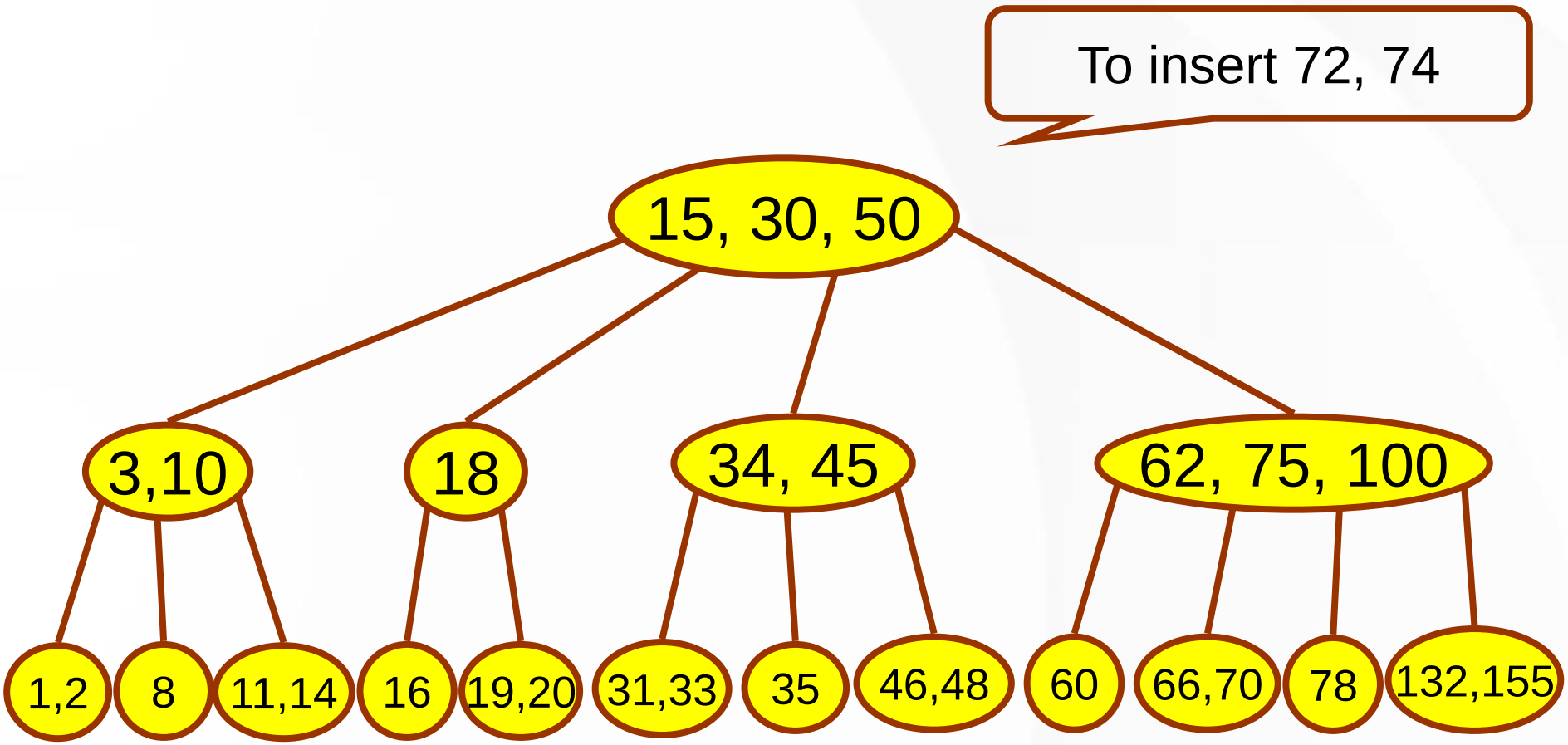


to insert 20

Illustration – Order of 3

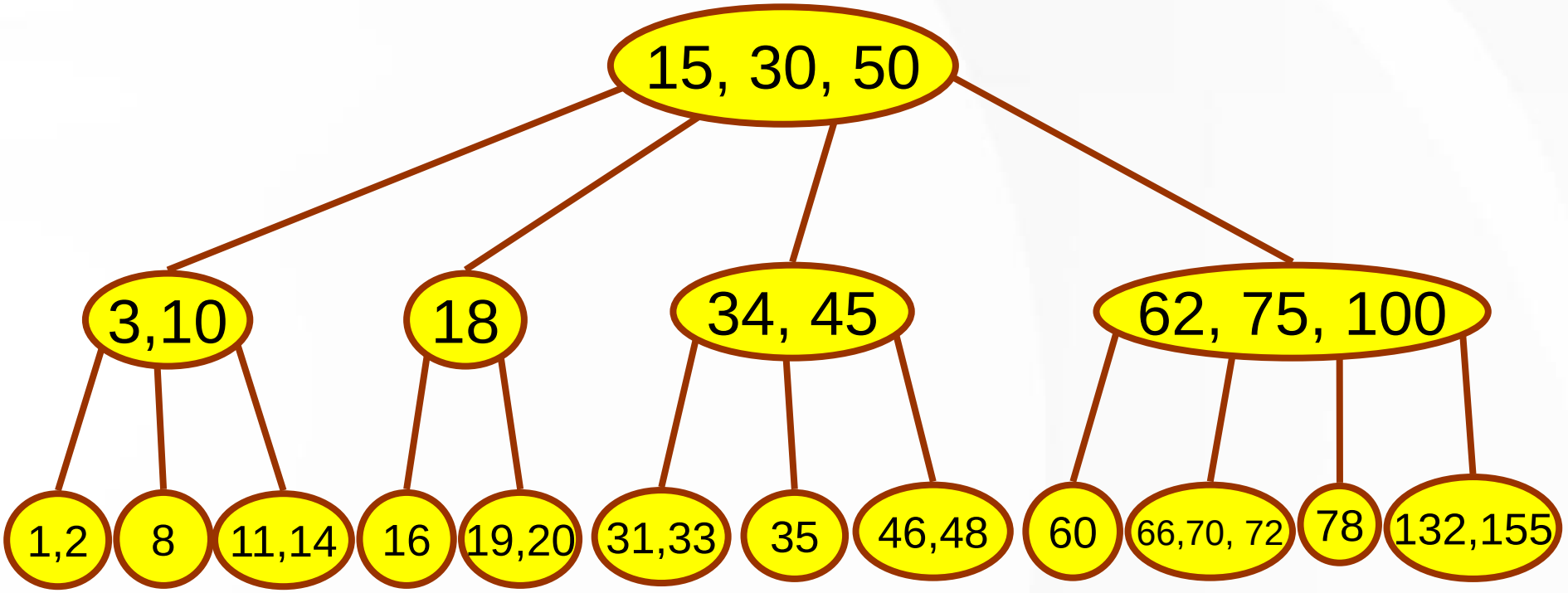


B-Tree (Order of 4)

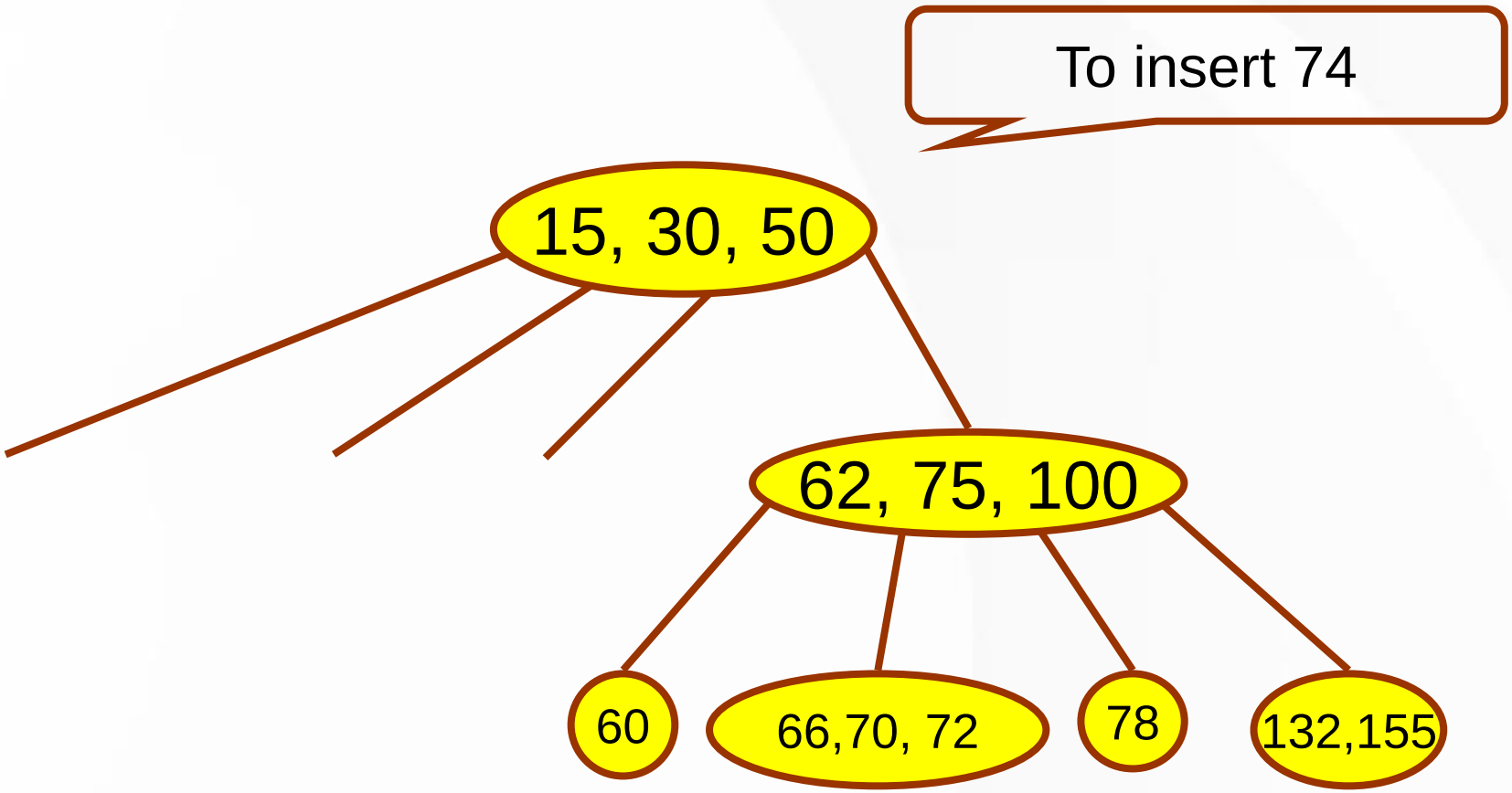


B-Tree (Order of 4)

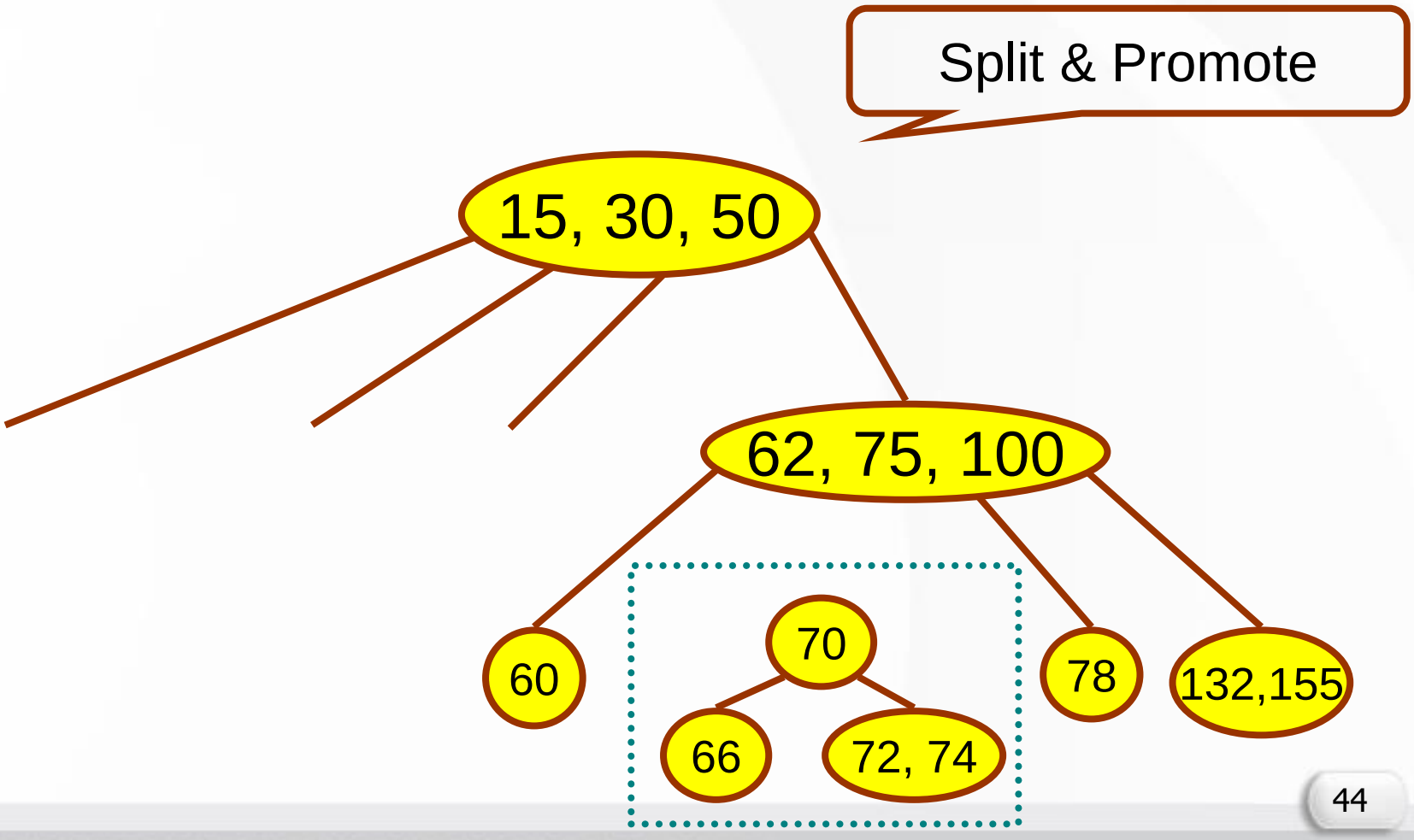
After inserting 72
To insert 74



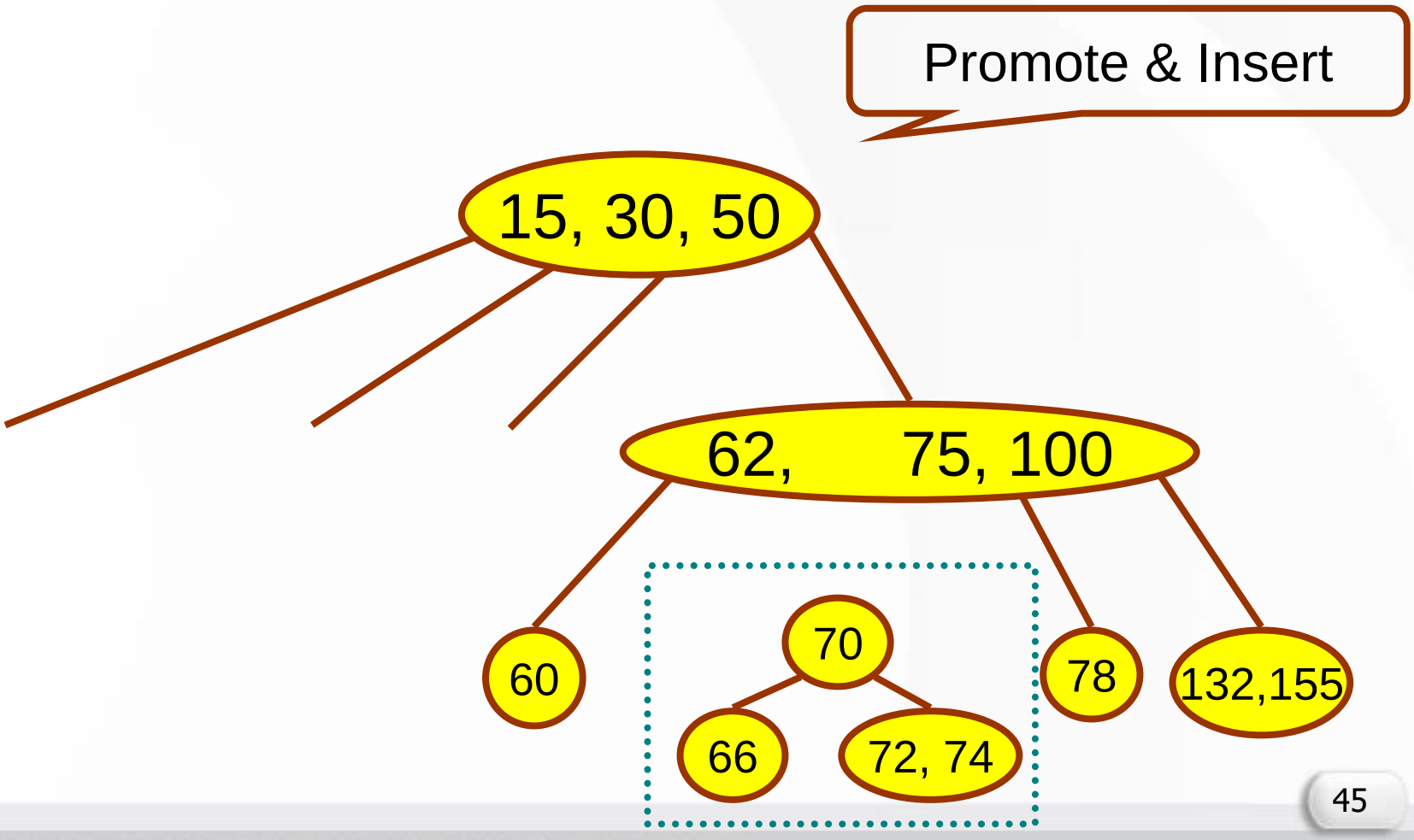
B-Tree (Order of 4)



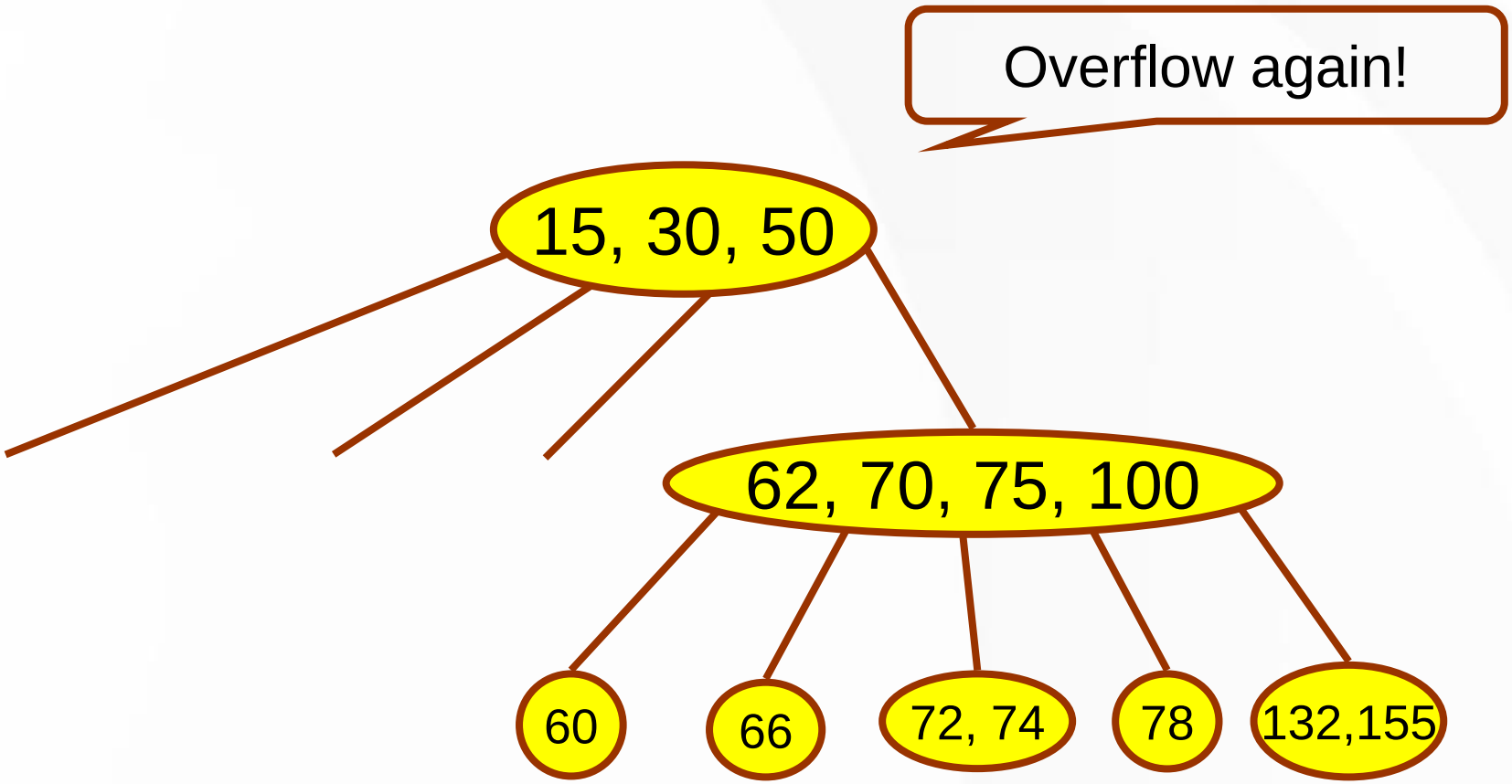
B-Tree (Order of 4)



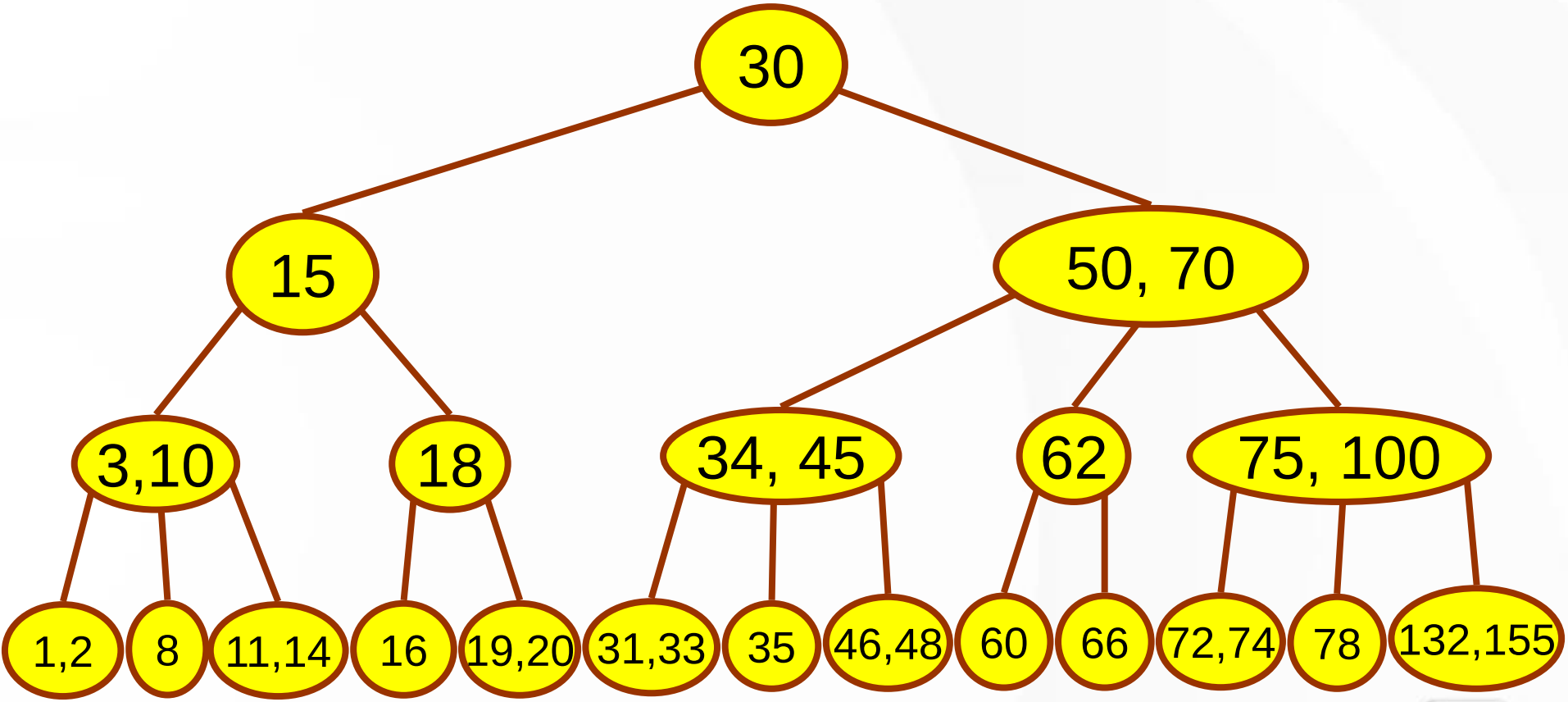
B-Tree (Order of 4)



B-Tree (Order of 4)



B-Tree (Order of 4)



Key Removal

Concept from BST

- Removing a node
  - Three cases
    - Leaf node
    - One subtree
    - Two subtrees
- Difference
  - Removing a key
  - Removing a node

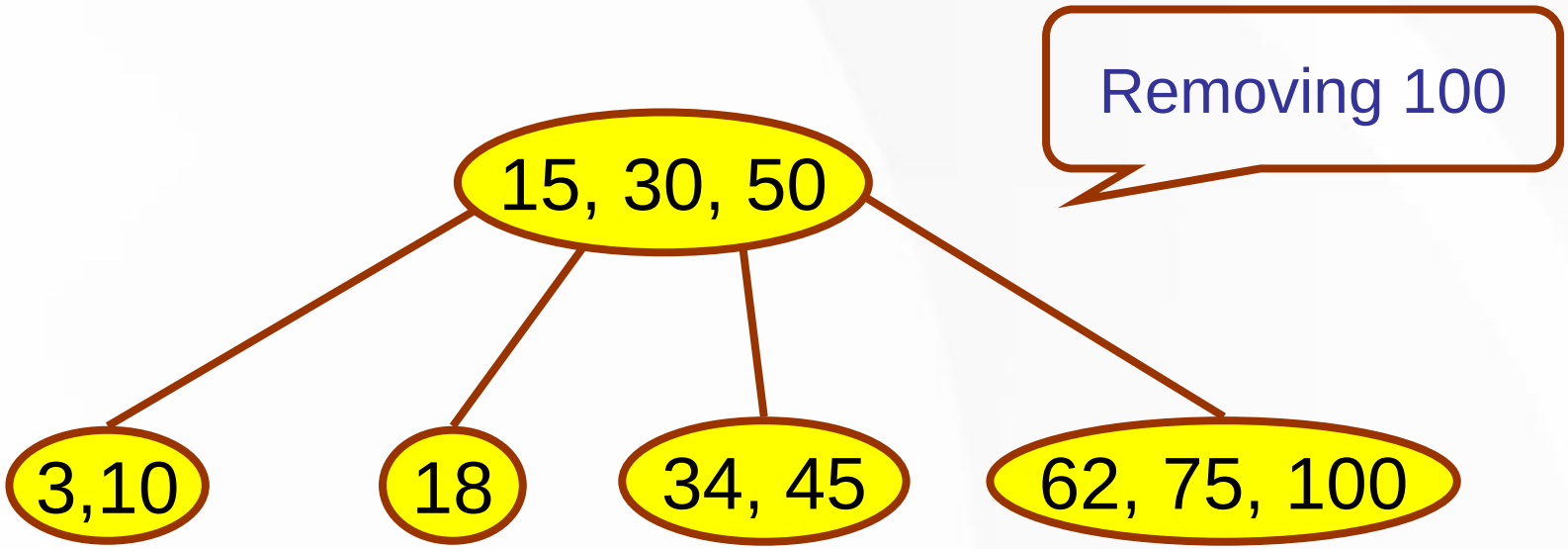
Procedures

- Search for a key
  - Start from the root
  - Sequential search in the root
  - If found, remove the key
  - If not found
    - Repeat for the key in the subtree
- Challenge
  - How to keep the requirements of “order”???
  - To avoid underflow

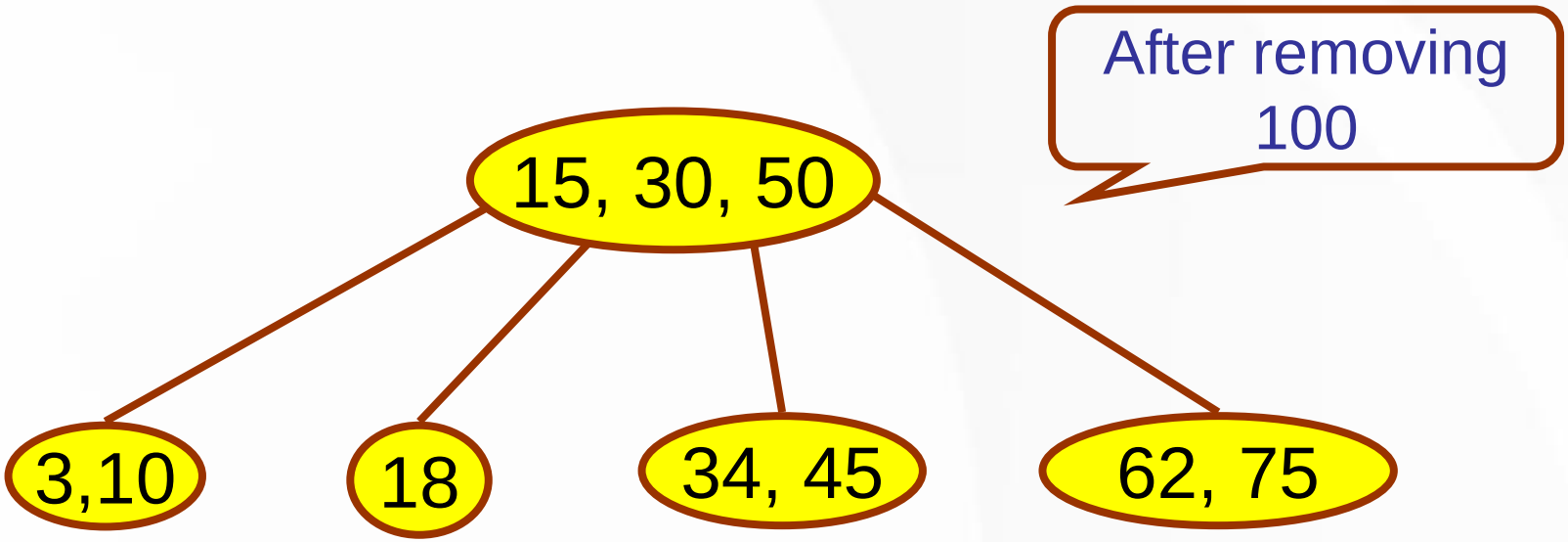
Order of B-Trees (number of keys)

Order	Number of Subtrees		Number of Keys	
	Minimum	Maximum	Min	Max
3	2	3	1	2
4	2	4	1	3
5	3	5	2	4
6	3	6	2	5
…	…	…	…	…
m	$\lceil (m/2) \rceil$	m	$\lceil (m/2) \rceil - 1$	m - 1

Simplest Procedure

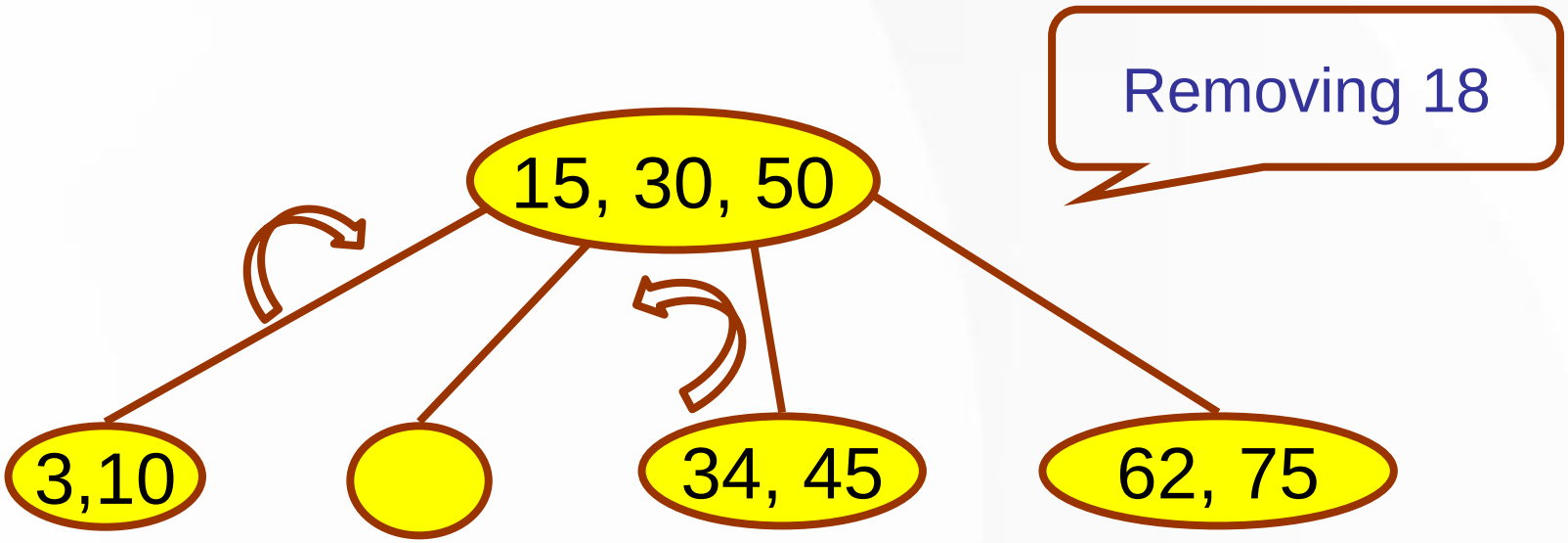


Simplest Procedure



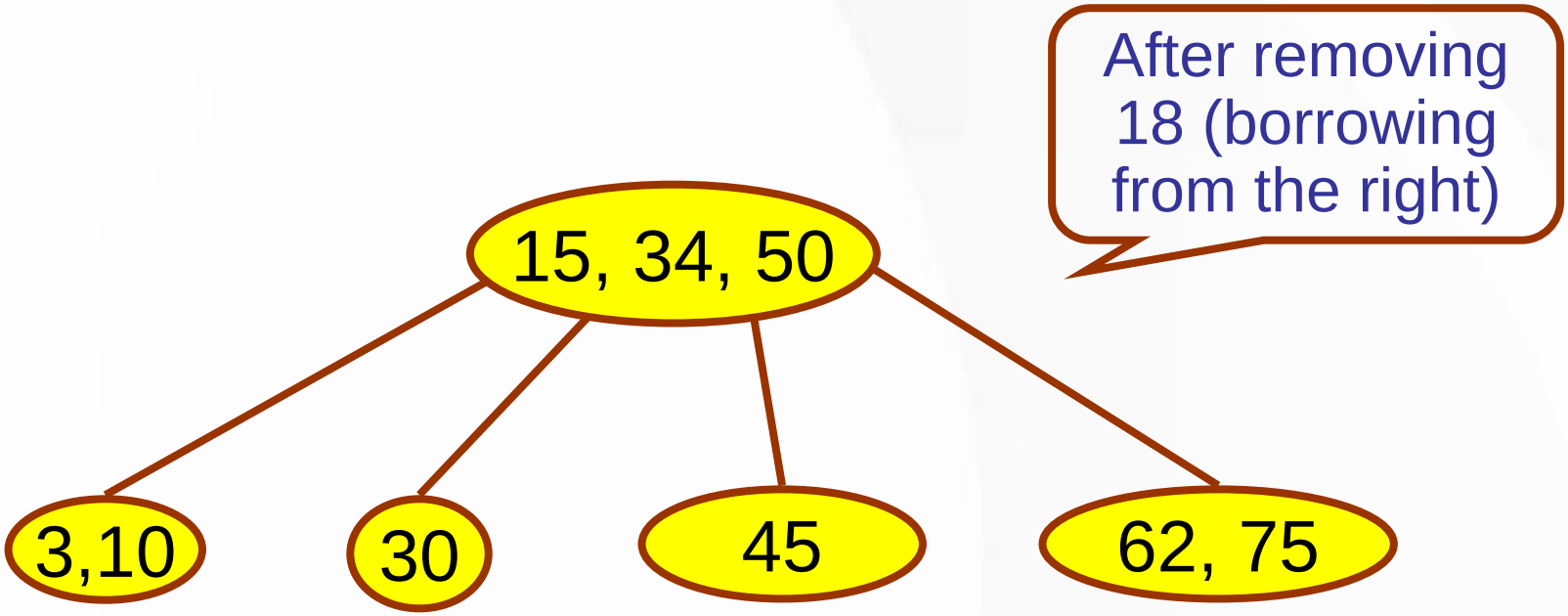
Borrowing Procedure

- How to avoid underflow?



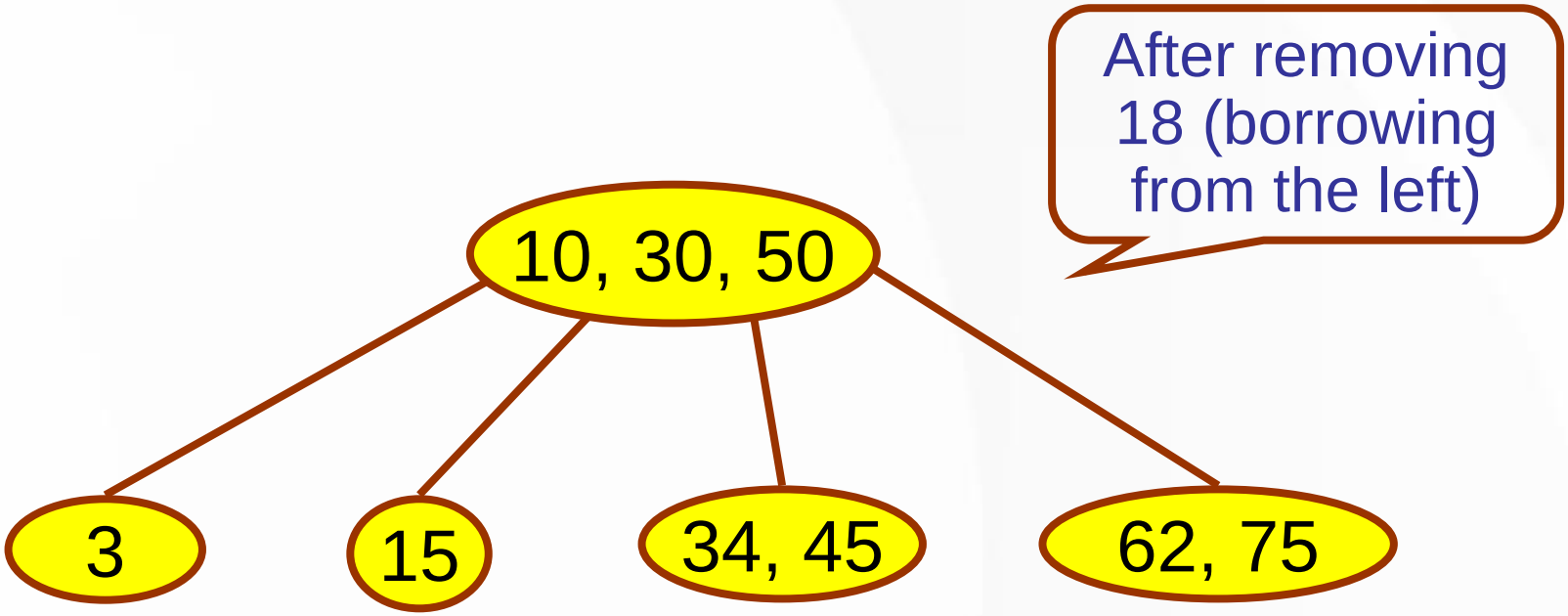
Borrowing Procedure

- How to avoid underflow?



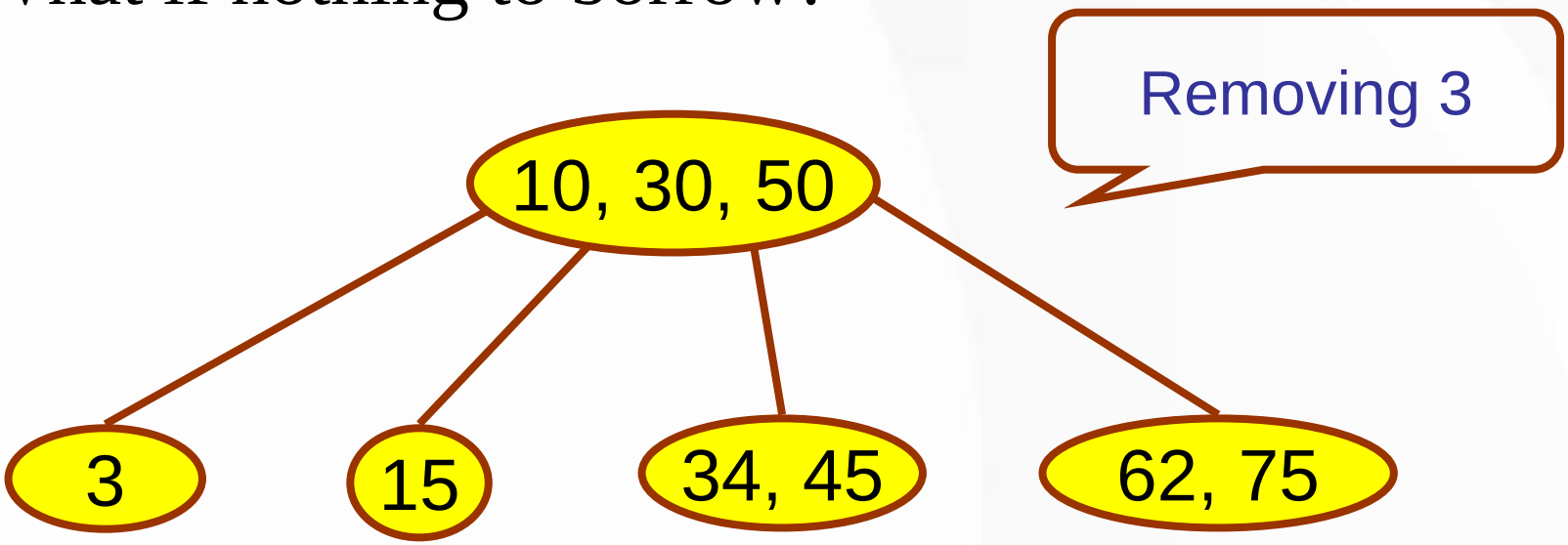
Borrowing Procedure

- How to avoid underflow?



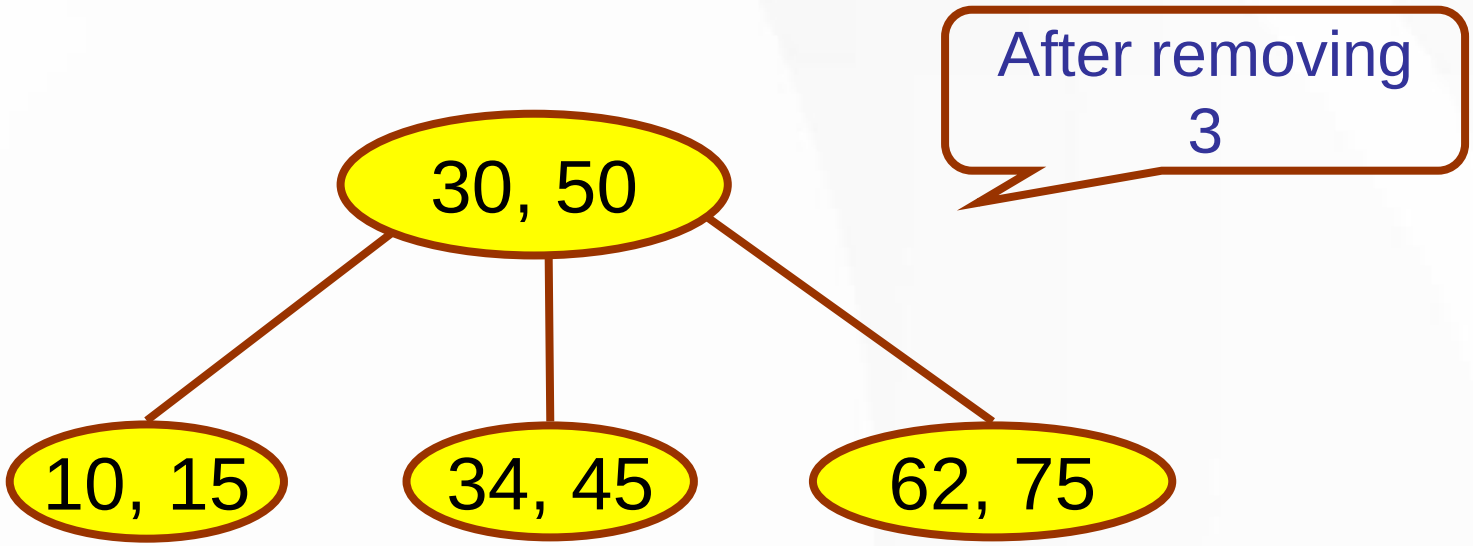
Combining Procedure

- How to avoid underflow?
- What if nothing to borrow?

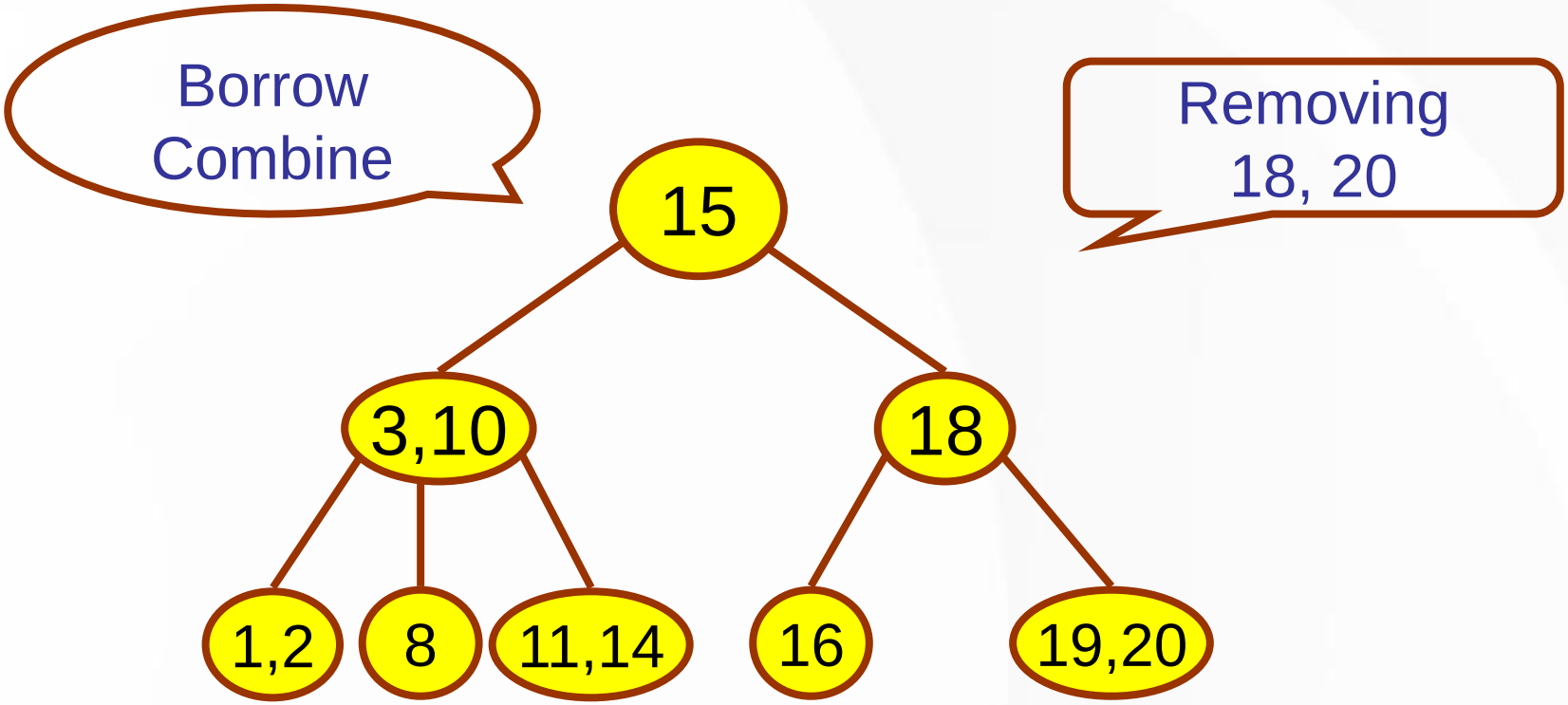


Combining Procedure

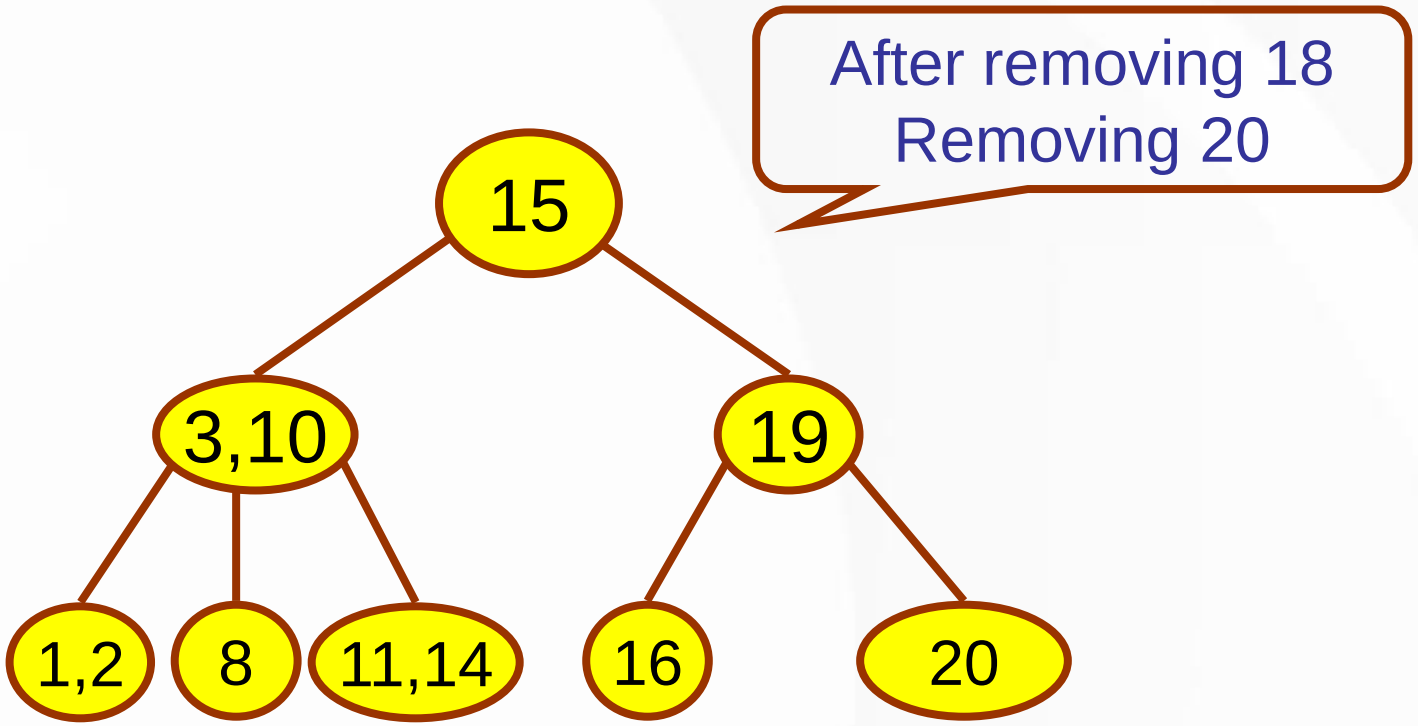
- Demote and combine (collapsing procedure)



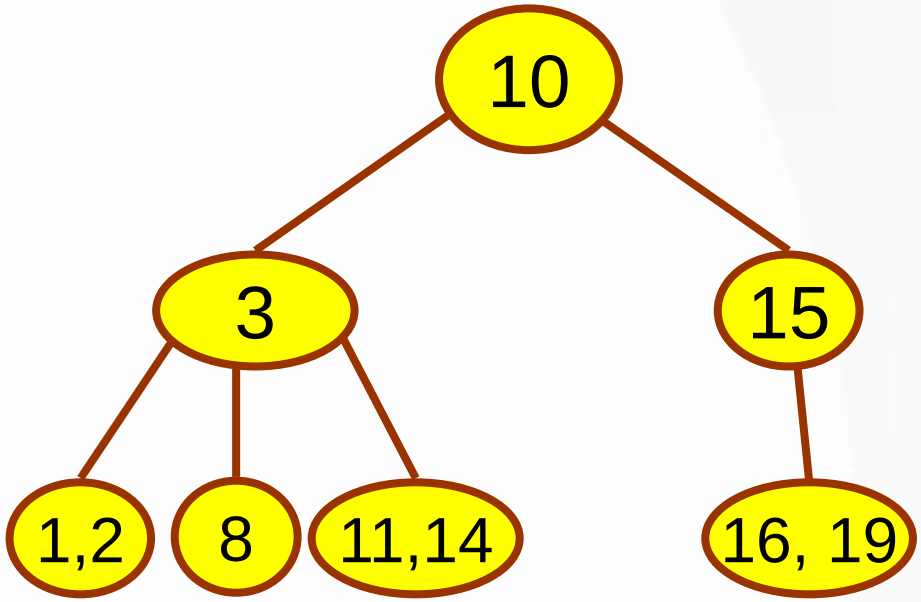
Combined Procedure



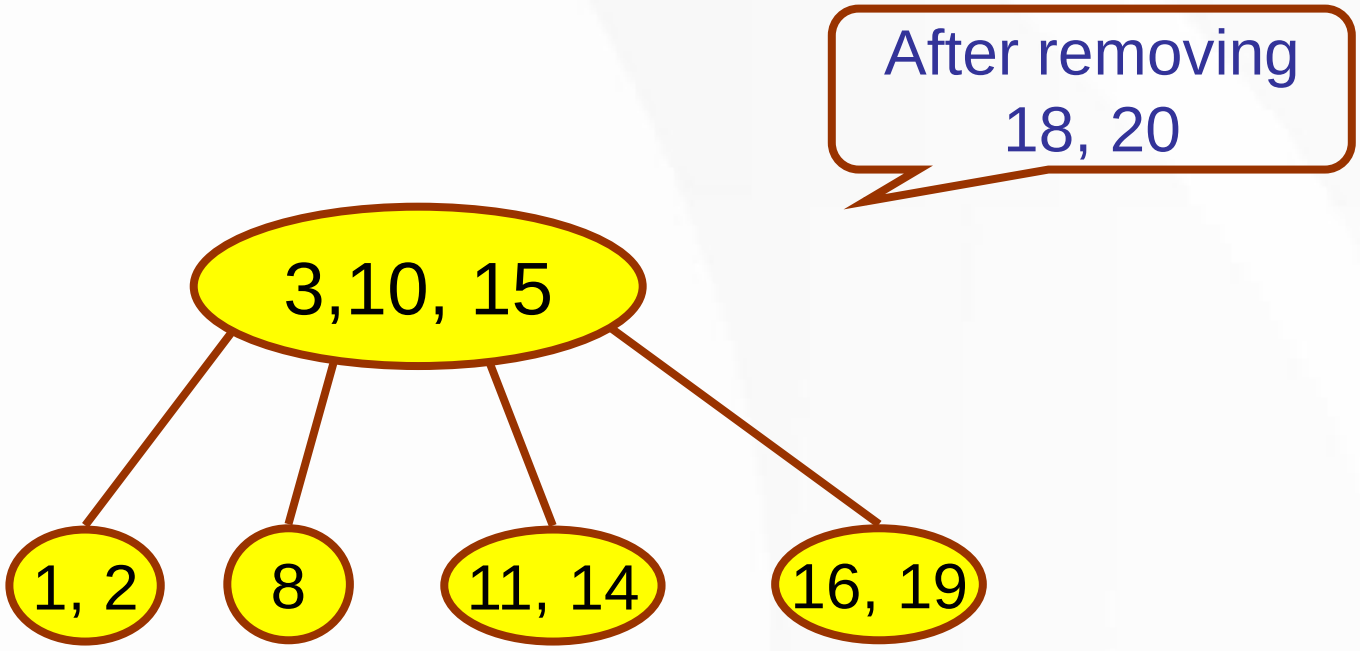
Combined Procedure



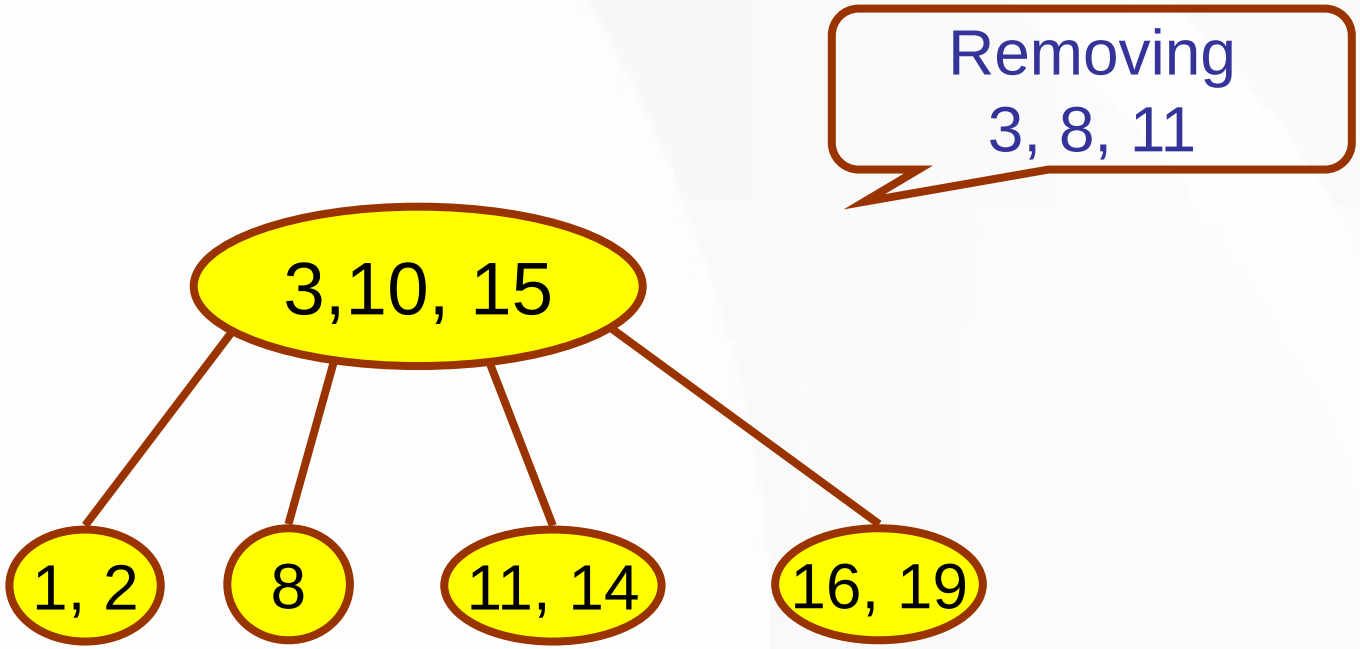
Combined Procedure



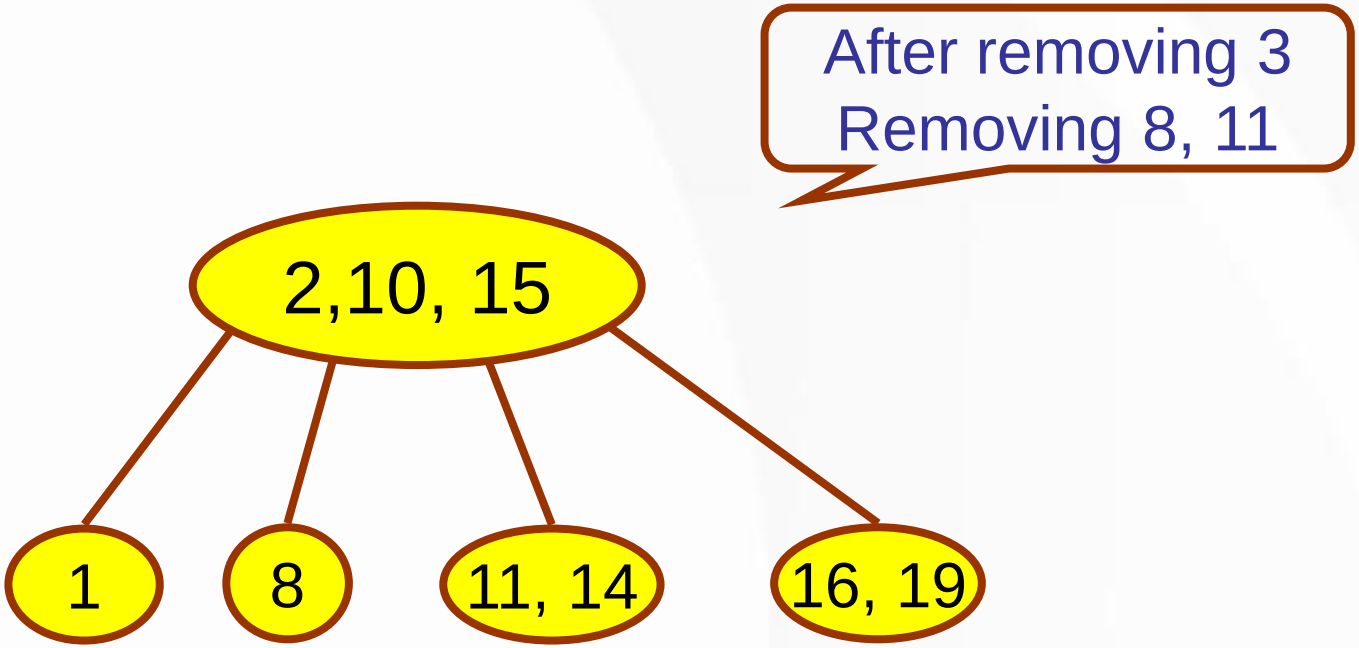
Combined Procedure



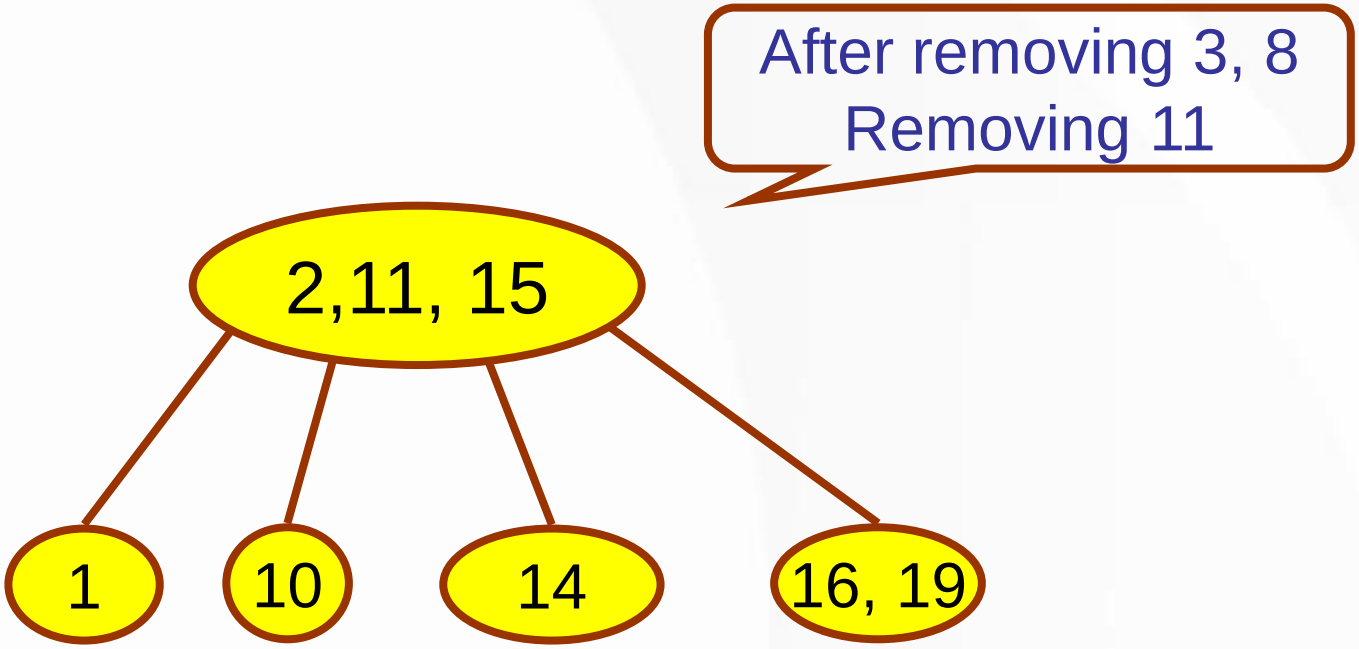
Combined Procedure



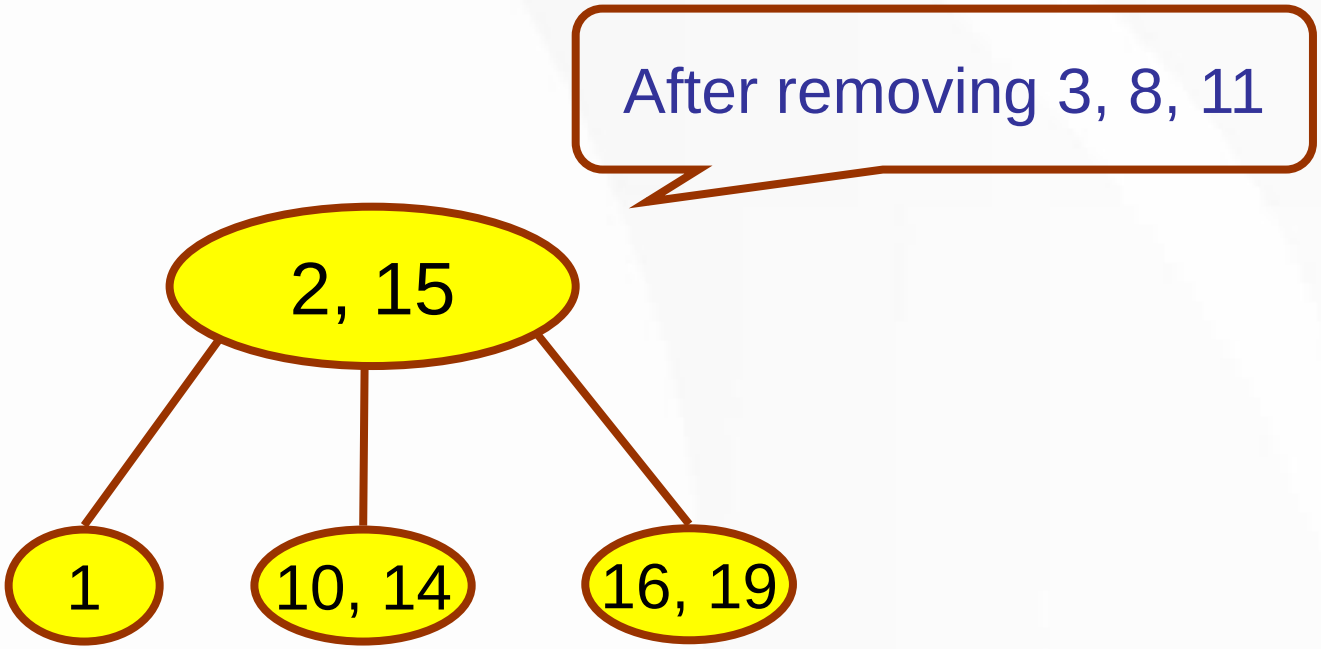
Combined Procedure



Combined Procedure



Combined Procedure



Variations

- 2-3 Tree
- B* Tree
- B+ Tree
- B# Tree

Order	Number of Subtrees		Number of Keys	
	Min	Max	Min	Max
3	2	3	1	2
4	2	4	1	3
5	3	5	2	4
6	3	6	2	5
…	…	…	…	…
m	(m/2)	m	(m/2) - 1	m - 1

Final Remarks

- Multiway trees
  - General concept
  - Shortcoming
- B-Trees
  - Insert
  - Remove
- Variation