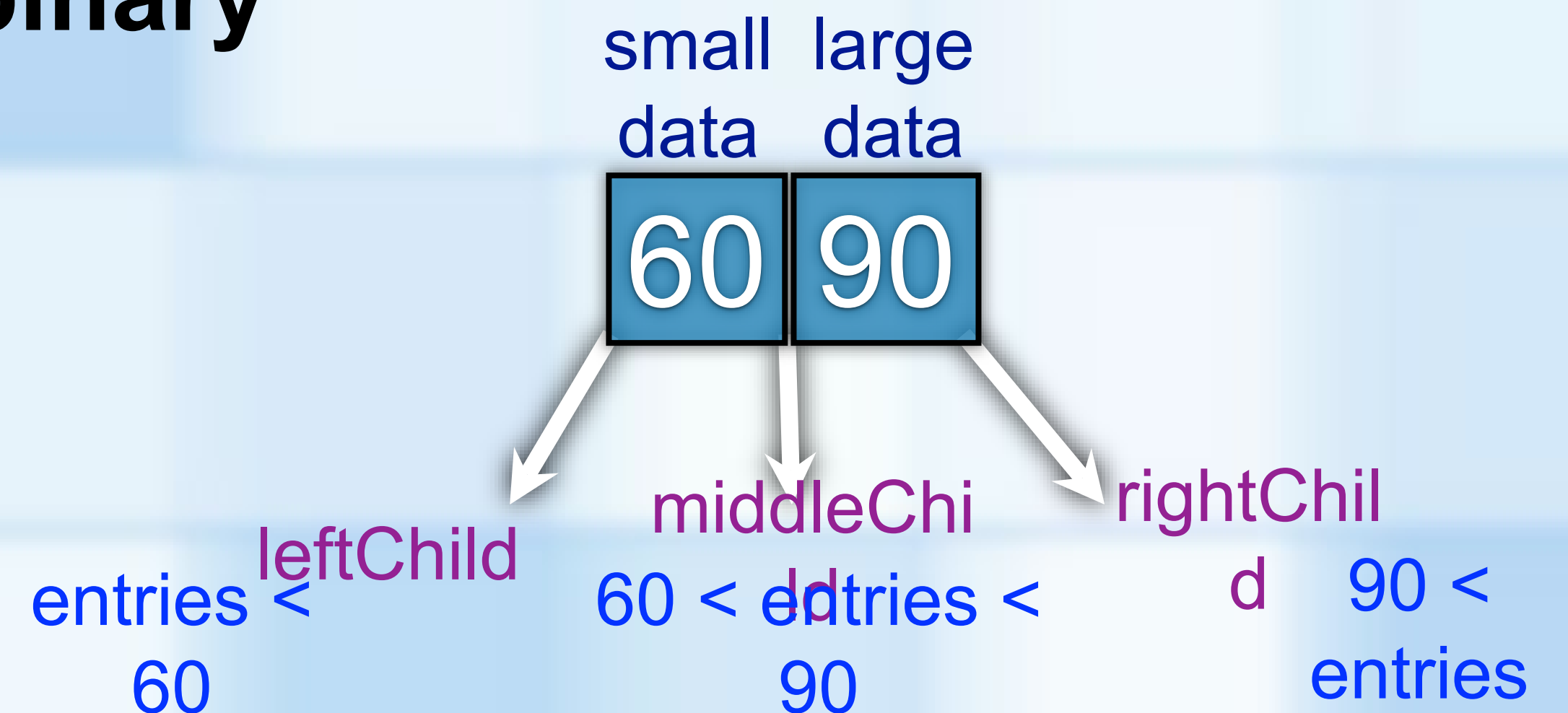
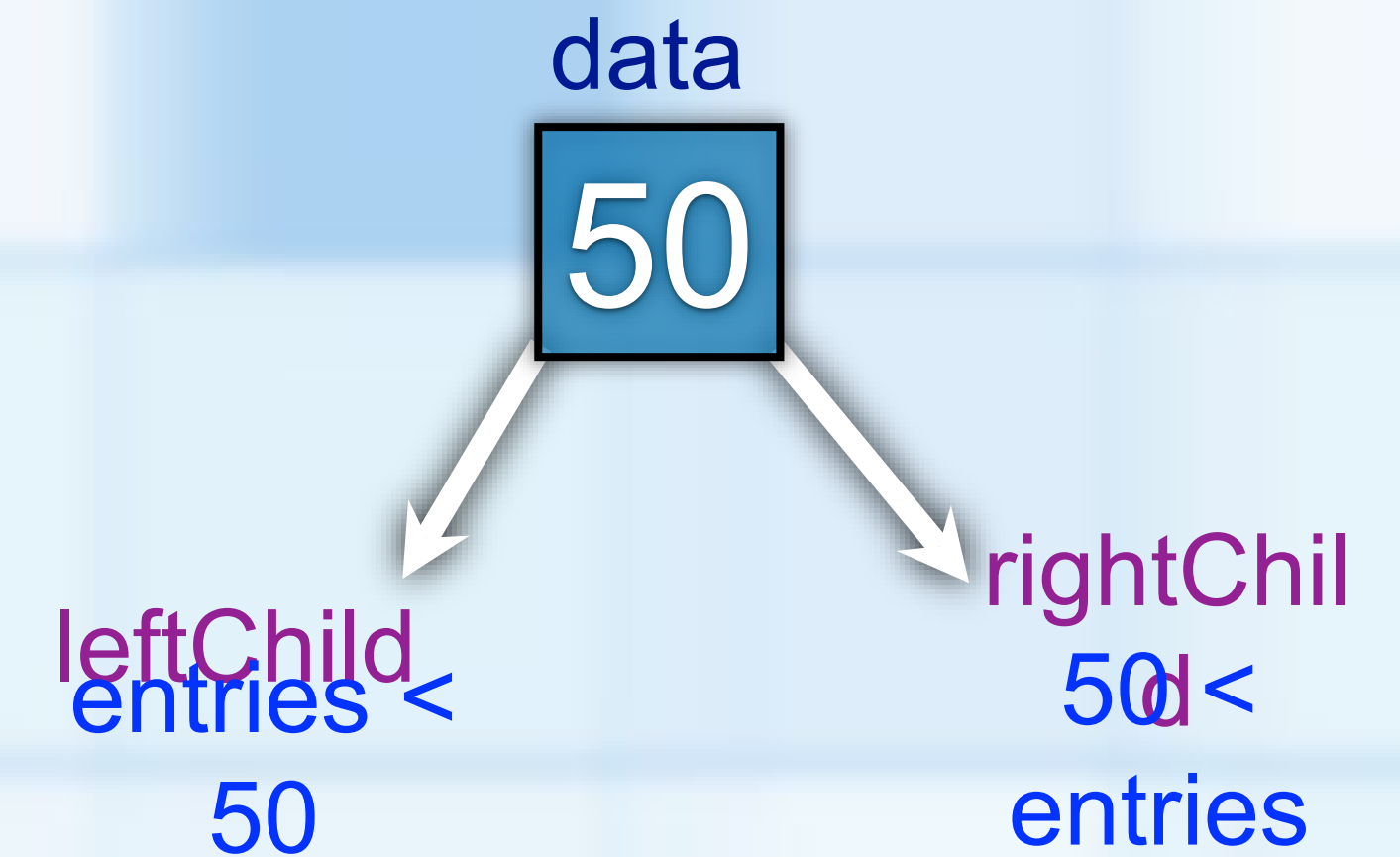


2-3 TREES

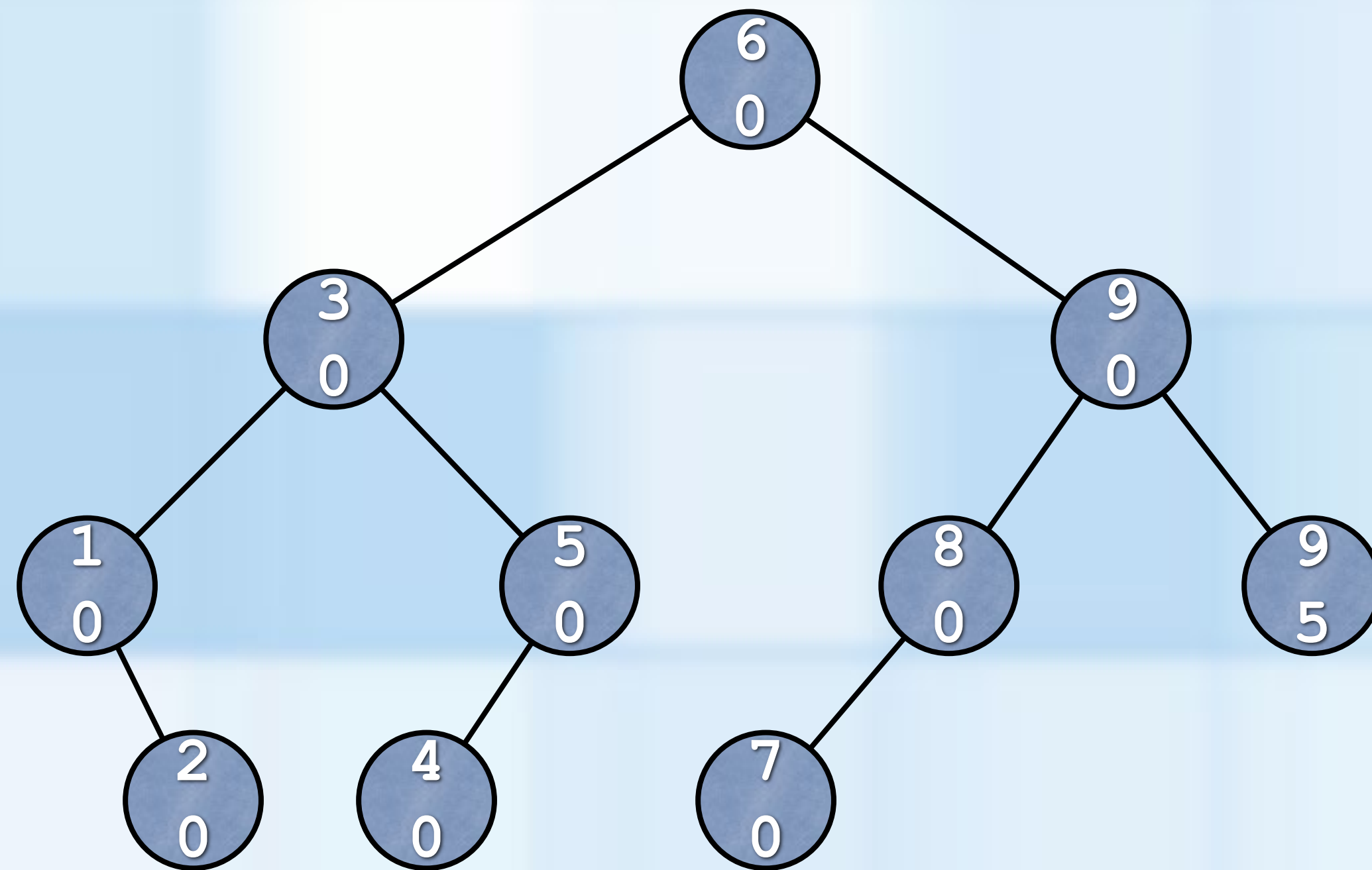
2-3 TREES

- General Search Tree
- Interior nodes are either **2-nodes** or **3-nodes**
 - **2-node** has one data item and two children
 - **3-node** has two data items and three children
 - Simple implementations use 3-node to represent both
- Are never taller than minimum-height binary tree
 - A 2-3 tree with n nodes never has height greater than $\log_2(n+1)$
- All leaves are at the same level
- 2-3 trees are not binary trees



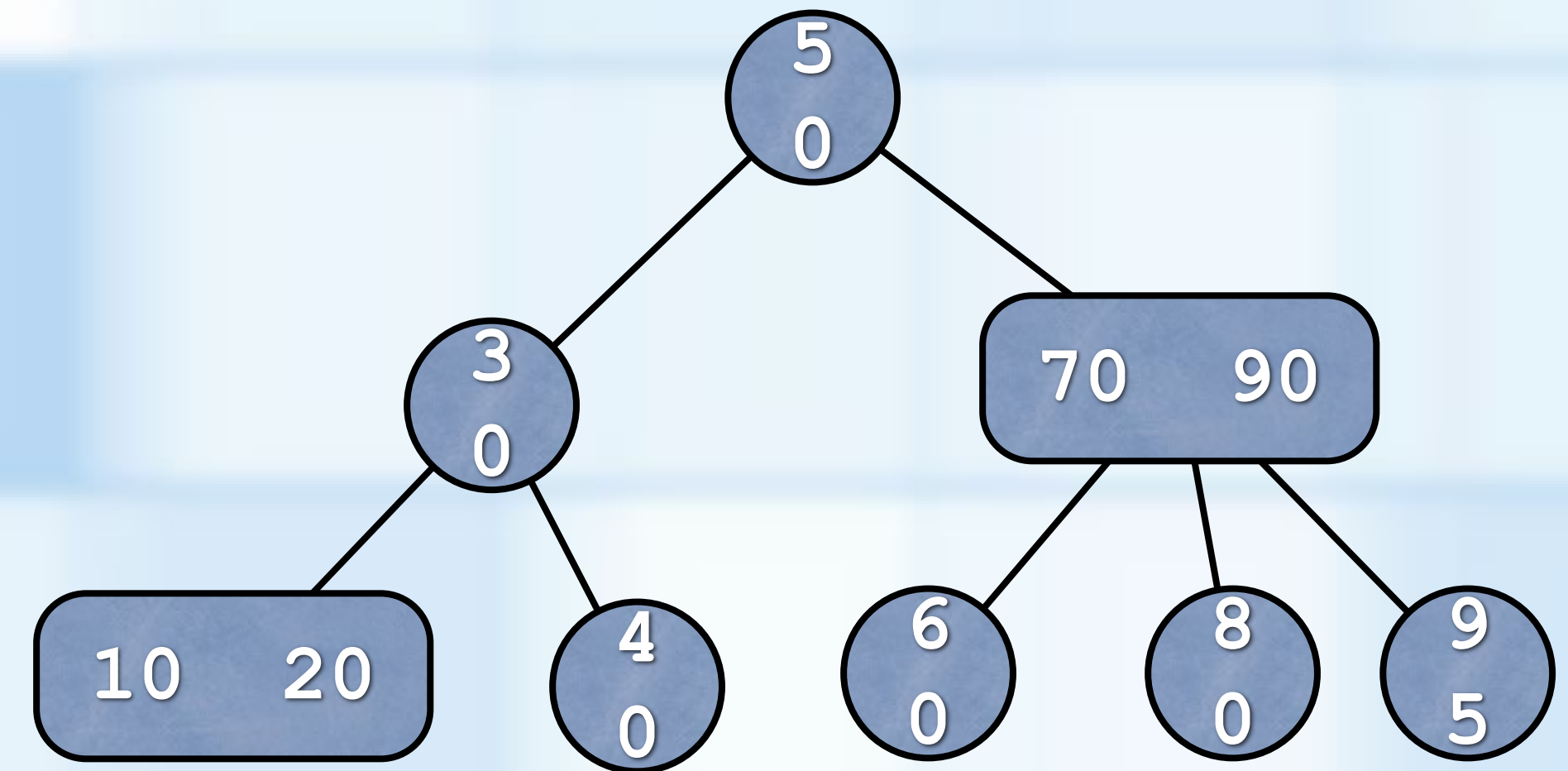
2-3 TREE HEIGHT

A balanced binary search tree



Height 4

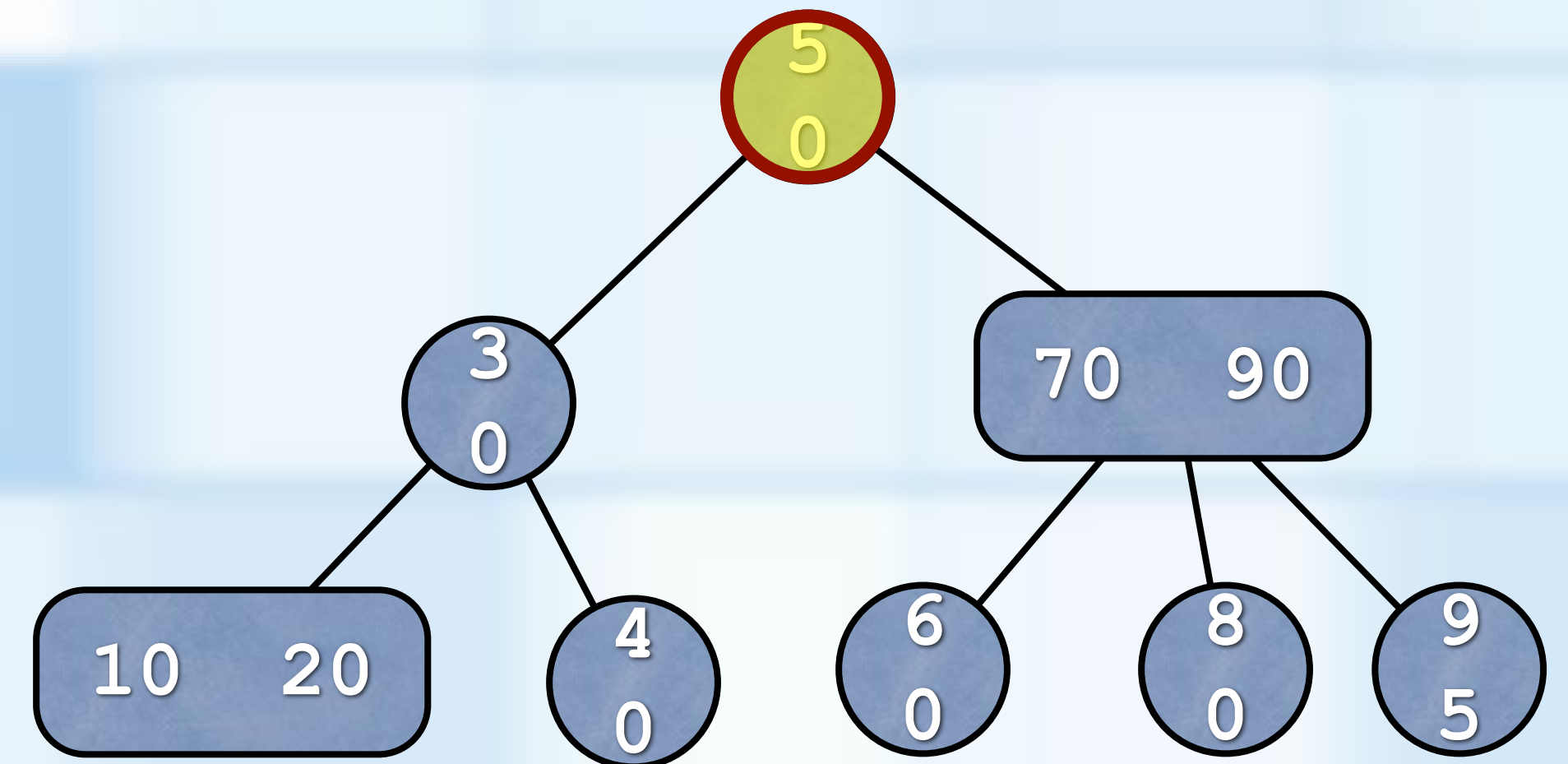
2-3 tree with the same elements



Height 3

TRAVERSING 2-3 TREES

- To traverse a 2-3 tree
 - Perform the analogue of an *in-order* traversal
 - leftmost subtree,
 - left value,
 - center subtree,
 - right value,
 - rightmost subtree
- Searching a 2-3 tree is as efficient as searching the shortest binary search tree

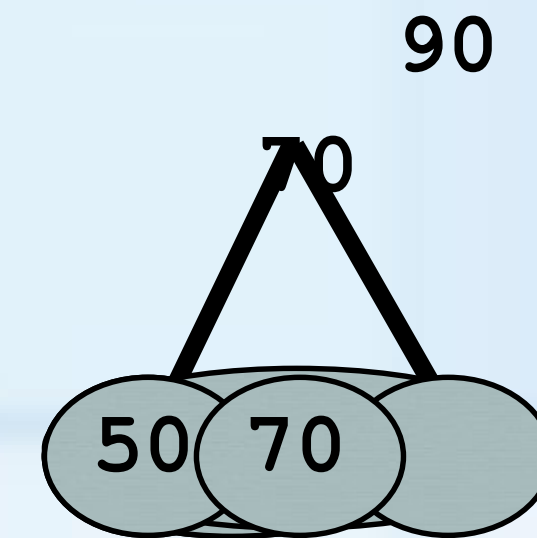


2-3 TREE OPERATIONS

INSERTING TO 2-3 TREES

- *Always insert values into an EXISTING leaf*
 - Only exception — first value inserted into empty tree
 - Create a 2-node and insert value
- Inserting a value into a 2-node turns it into a 3-node
- Inserting a value into a 3-node causes it to divide
 - Result is subtree of three 2-nodes

50 70 90

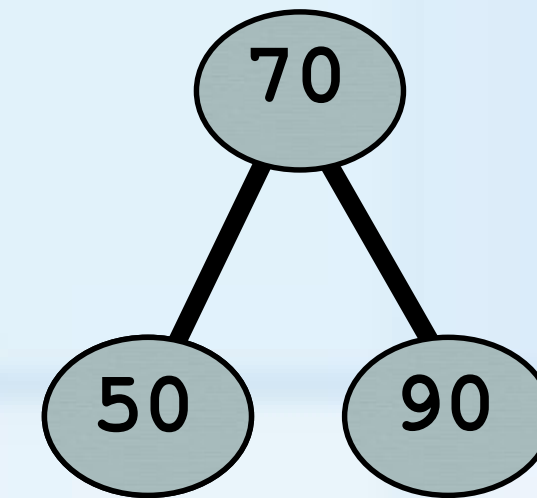


INSERTING TO 2-3 TREES

- To insert an item with search key k into a 2-3 tree

50 70 90

- Locate the leaf at which the search for k would terminate
- Insert the new item k into the leaf
- If the leaf now contains only two items, you're done
- If the leaf now contains three items,
 - split the leaf into 2-nodes and move middle value into parent node
 - If no parent node exists,



INSERTING TO 2-3 TREES

- When an internal node contains three items

- Split the node into two nodes
- Accommodate the node's children

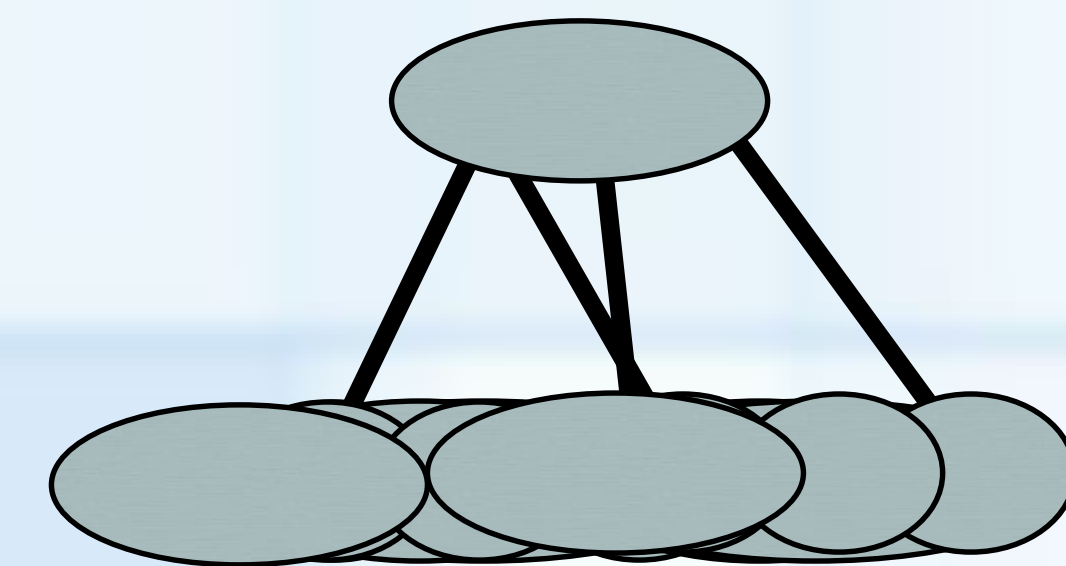
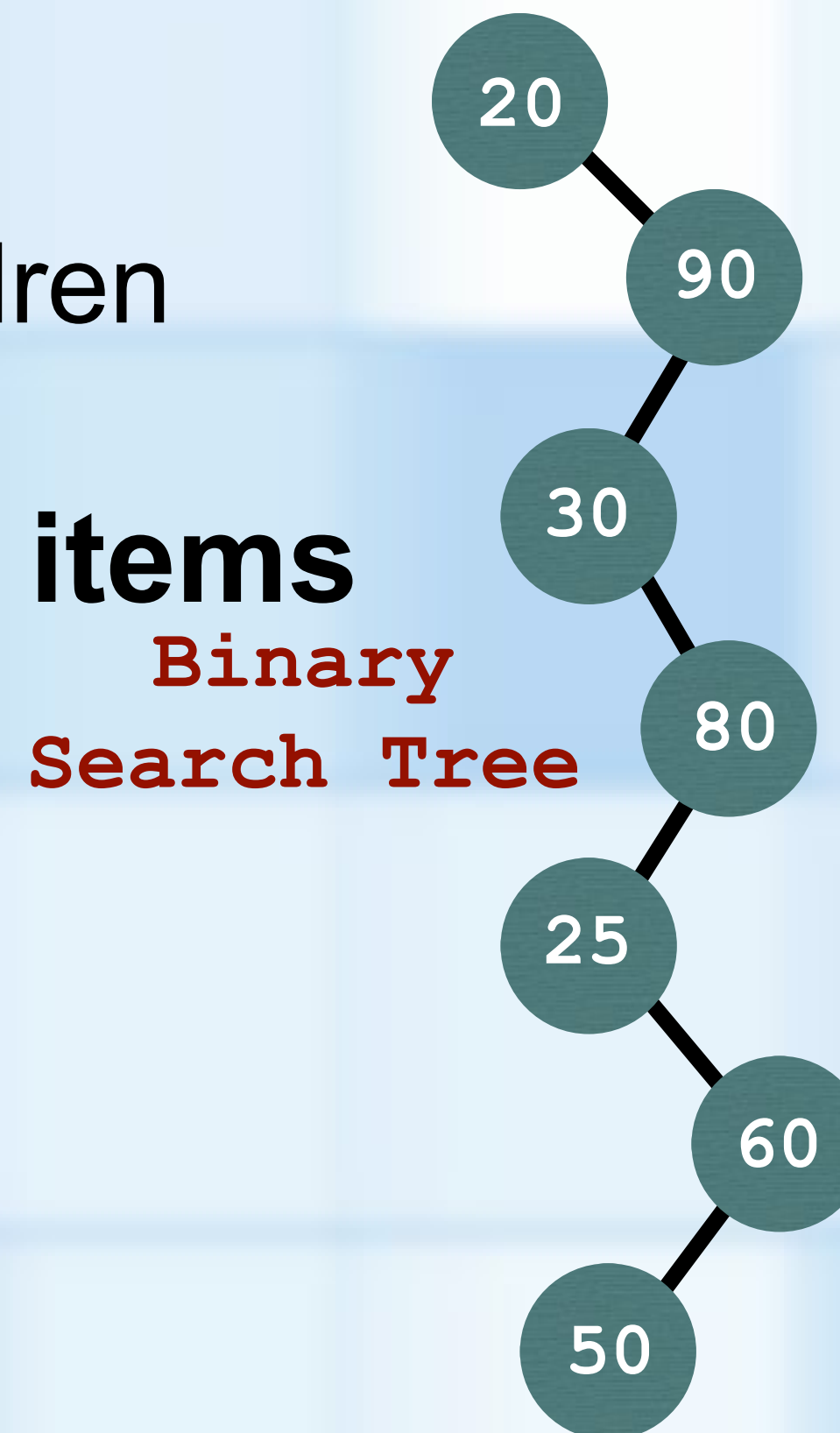
- When the root contains three items

- Split the root into two nodes
- Create a new root node

- Tree grows in height

Order Values Given:

20 90 30 80 25 60 50



2-3 Tree

REMOVING FROM A 2-3 TREE

- Removing values from 2-3 trees
 - Always remove value from a leaf
 - **Other values** (and children) are redistributed
 - Nodes can be merged
 - **Root node is only node ever removed**
 - and only if it is empty (contains no values)

