COL106: Data Structures and Algorithms

Ragesh Jaiswal, IIT Delhi

- Question: How do we perform `remove(k)` operation on an AVL tree?

Algorithm Sketch

//Initially p denotes the parent of the removed node

`BalanceAfterRemove(Node` p`)`

- Let z be the first unbalanced node going up from p
- If no such z exists then return
- Let y be the child of z of greater height.
- Let x be the child of y defined as follows:
  If one child of y is taller than the other then x is the taller child, otherwise x is the child of y with the same side as y is of z
- Perform Tri-node restructuring w.r.t. x, y, z
- Let b denote the tallest node (among the nodes involved in restructuring) after the restructuring.
- If b is not the root, then `BalanceAfterRemove(b.parent)`

Data Structures: Balanced ~~Binary~~ Search Trees

- Binary search trees allows storage and access of data in time proportional to the height of the tree.
- The number of children of a node binary search trees is upper bounded by 2.
- Removing this restriction might provide us more flexibility without costing us in terms of performance.
- Multiway Search Trees are generalisation of Binary Search Trees where internal nodes are allowed to have more than two children.

- Multiway Search Trees are generalisation of Binary Search Trees where internal nodes are allowed to have more than two children.
- A node in an ordered tree is said to be a d-node iff it has d children.

Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

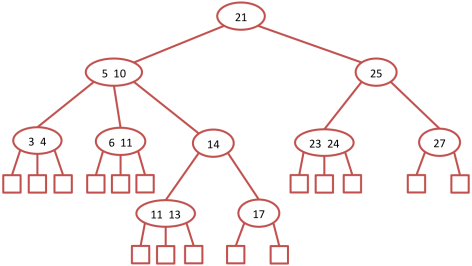
- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree?

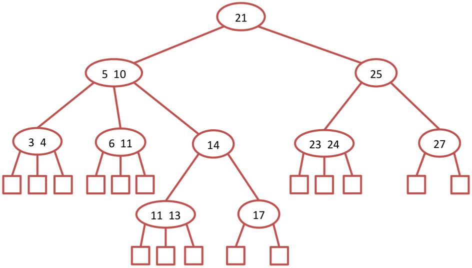


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree? No

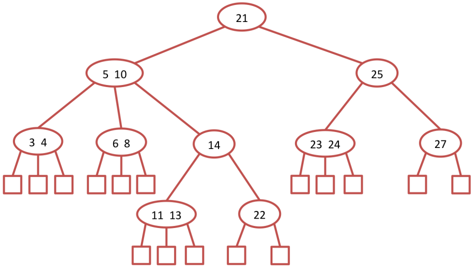


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree?

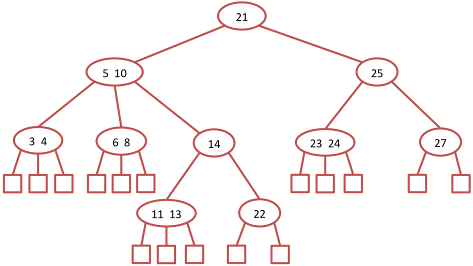


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree? No

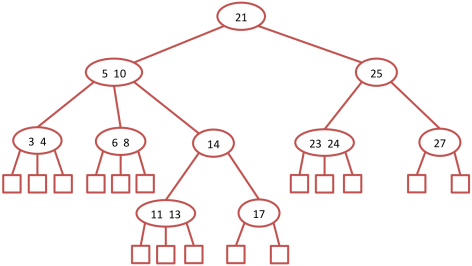


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree?

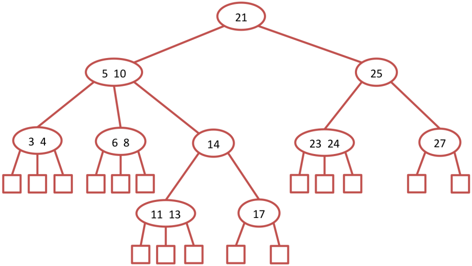


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Is this a Multiway Search Tree? Yes



Data Structures

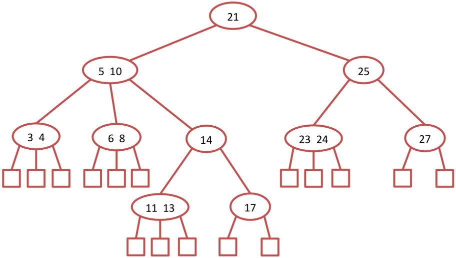
~~Binary~~ Search Trees $\rightarrow$ Multiway Search Trees

Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- <u>Claim 1</u>: Any multiway search tree containing n entries has $n + 1$ leaves.

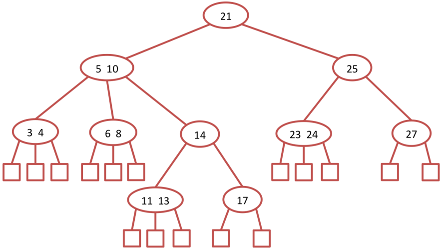


Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

- Question: How do we perform search operation on multiway search trees?
  - How do we search for an entry with key 17 in the tree below?



Definition (Multiway Search Tree)

A Multiway Search Tree is an ordered tree that has the following properties:

- Each internal node is a d-node with $d \geq 2$.
- Each internal d-node with children $c_1, ..., c_d$ stores an ordered set of $(d-1)$ key-value pairs $(k_1, v_1), (k_2, v_2), ..., (k_{d-1}, v_{d-1})$, where $k_1 \leq k_2 \leq ... \leq k_{d-1}$.
- Consider any d-node w. Let $k_0 = -\infty$ and $k_d = +\infty$ by convention. For every entry (k, v) stored in the subtree rooted at c_i, we have $k_{i-1} \leq k < k_i$.

Definition ((2-4)-Tree)

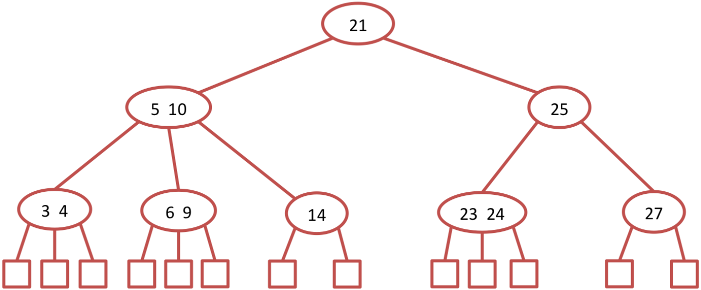
A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.



Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Claim 1: The height of any (2, 4)-tree storing n entries is $O(\log n)$.

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Claim 1: The height of any (2, 4)-tree storing n entries is $O(\log n)$.
- Question: What is the running time of search operation on (2, 4)-Tree?

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Claim 1: The height of any (2, 4)-tree storing n entries is $O(\log n)$.
- Question: What is the running time of search operation on (2, 4)-Tree? $O(\log n)$

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

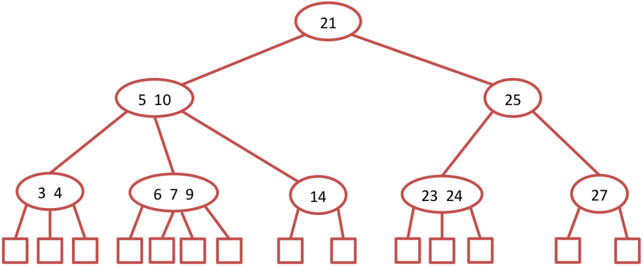
- Claim 1: The height of any (2, 4)-tree storing n entries is $O(\log n)$.
- Question: What is the running time of search operation on (2, 4)-Tree? $O(\log n)$
- Question: How do we perform an insert operation?

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?

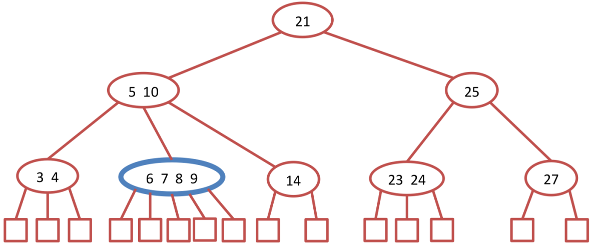


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?
  - The tree below is not a (2, 4)-Tree since it violates the size property. This violation at the blue node is called overflow.

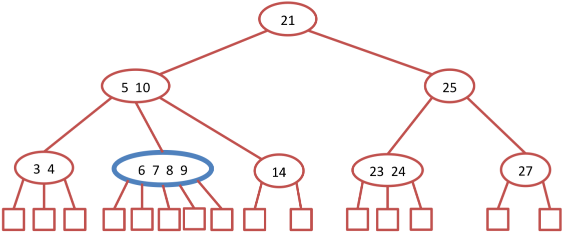


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?
  - The tree below is not a (2, 4)-Tree since it violates the size property. This violation at the blue node is called overflow.
  - Question: How do we resolve an overflow?

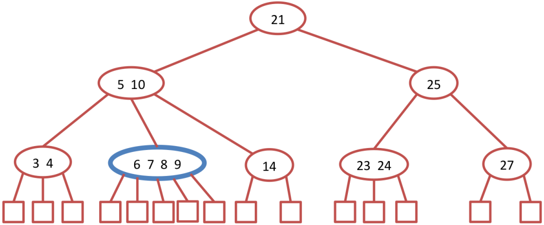


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?
  - The tree below is not a (2, 4)-Tree since it violates the size property. This violation at the blue node is called overflow.
  - Question: How do we resolve an overflow?
    - Perform a split at the overflow node.

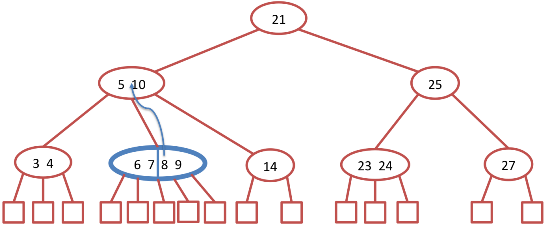


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?
  - The tree below is not a (2, 4)-Tree since it violates the size property. This violation at the blue node is called overflow.
  - Question: How do we resolve an overflow?
    - Perform a split at the overflow node.

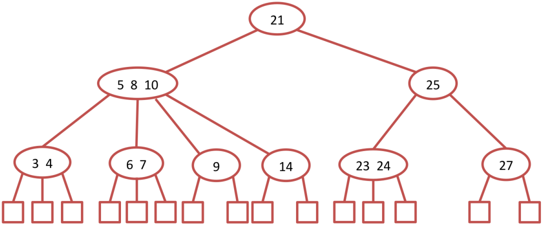


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform an insert operation?
  - How do we insert the key 8 in the tree below?
  - The tree below is not a (2, 4)-Tree since it violates the size property. This violation at the blue node is called overflow.
  - Question: How do we resolve an overflow?
    - Perform a split at the overflow node.

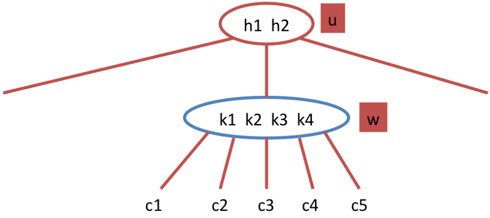


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Split operation on an overflow node *w*:

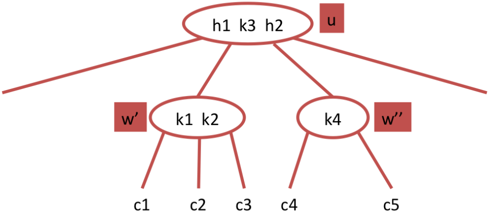


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Split operation on an overflow node.

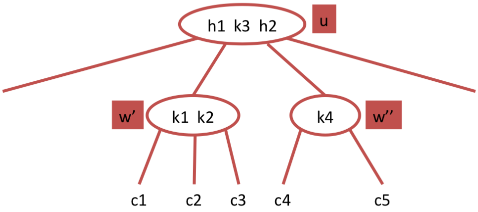


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Split operation on an overflow node.
- Complications:
  - What if the overflow node w is the root node?

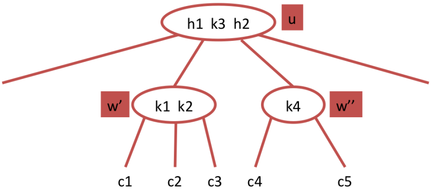


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Split operation on an overflow node.
- Complications:
  - What if the overflow node *w* is the root node? create a new root node
  - What if after the split, the node *u* overflows?

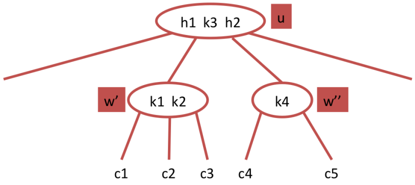


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Split operation on an overflow node.
- Complications:
  - What if the overflow node w is the root node? create a new root node
  - What if after the split, the node u overflows? continue performing split at u



Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: What is the running time of insert in (2, 4)-Tree?

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

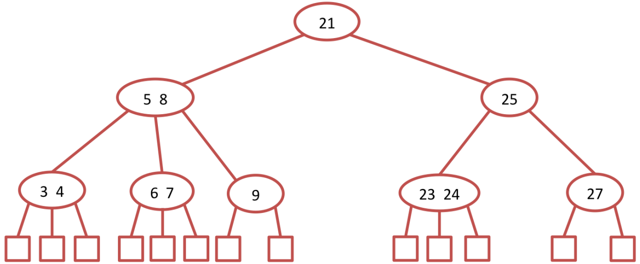
- Question: What is the running time of insert in (2, 4)-Tree?
  $O(\log n)$

Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - How do we delete the entry with key 8 from the tree below?

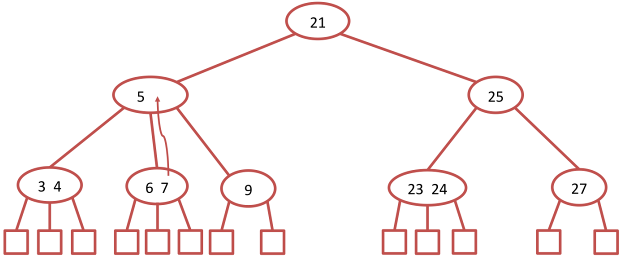


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - How do we delete the entry with key 8 from the tree below? Similar to BST

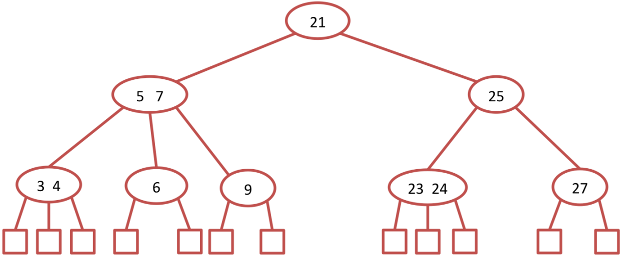


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - How do we delete the entry with key 8 from the tree below?
    Similar to BST

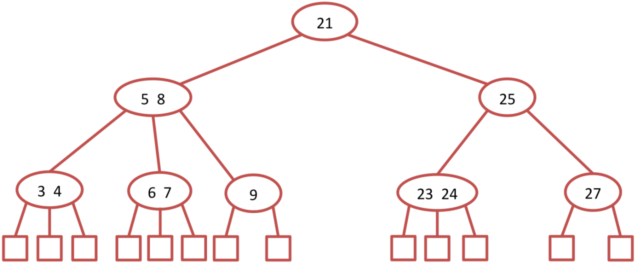


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - How do we delete the entry with key 27 from the tree below?

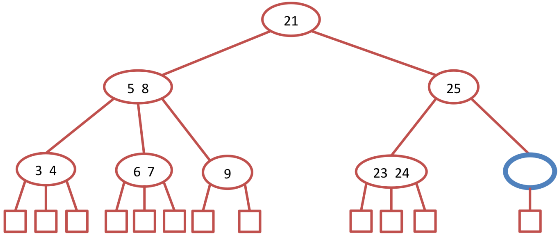


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - How do we delete the entry with key 27 from the tree below?
  - Since the children of node with key 27 are leaves, we can simply remove 27. This however creates a node with less than 2 children.
  - This condition is called underflow.

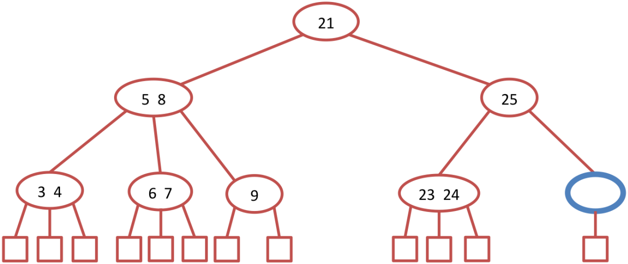


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node?

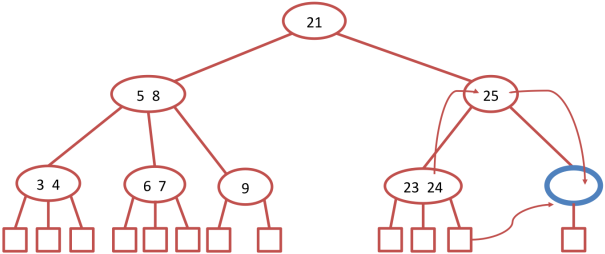


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node?
    - If possible, borrow an entry from a sibling.

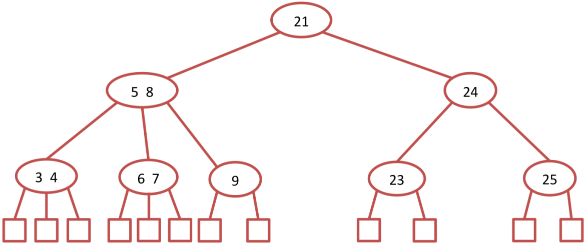


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible
  - Now consider deleting 21.

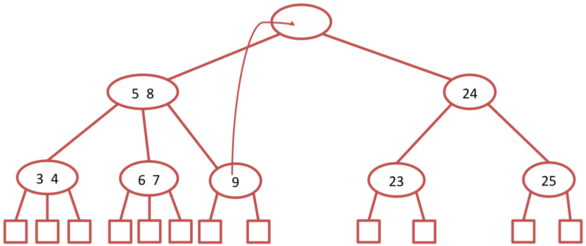


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible
  - Now consider deleting 21.

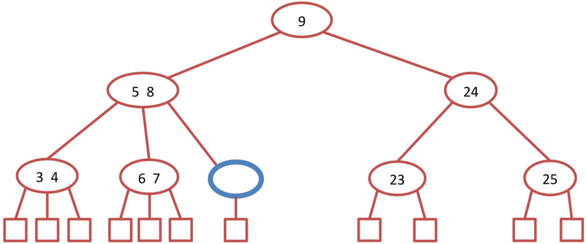


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible
  - Now consider deleting 21.

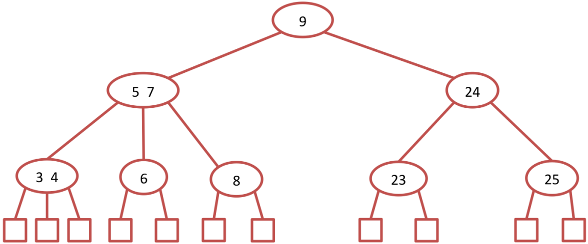


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
    - Question: How do we resolve underflow at a node? borrow from sibling if possible
    - Now consider deleting 8.

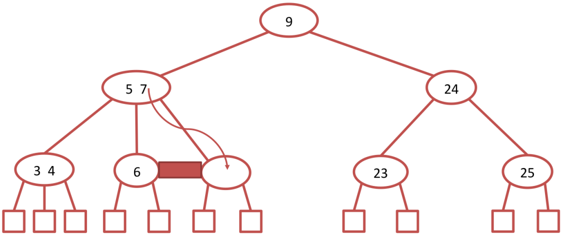


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible
  - Now consider deleting 8.
  - Since borrowing is not possible, perform fusion.

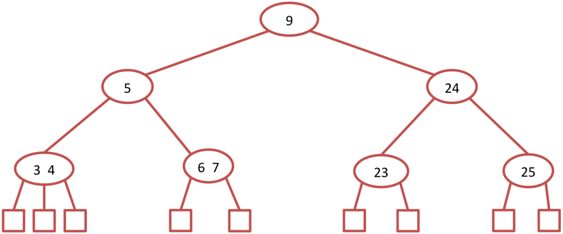


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible
  - Now consider deleting 8.
  - Since borrowing is not possible, perform fusion.

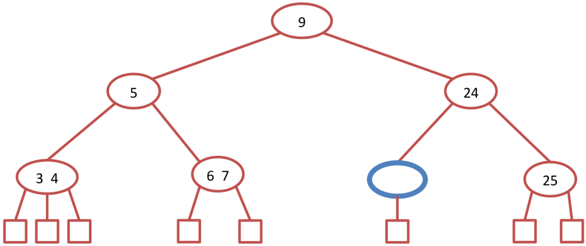


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible else perform fusion.
  - Now consider deleting 23.

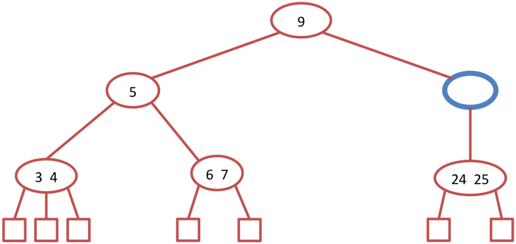


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible else perform fusion.
  - Now consider deleting 23.

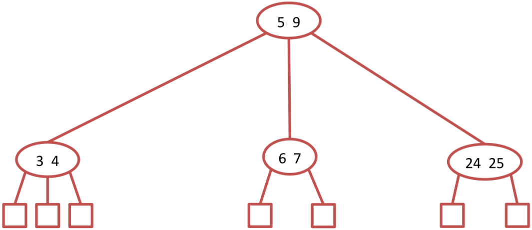


Definition ((2-4)-Tree)

A (2, 4)-Tree is a multiway search tree with the following two additional properties:

1. Size property: Every internal node has at most 4 children.
2. Depth property: All leaves have the same depth.

- Question: How do we perform deletion?
  - Question: How do we resolve underflow at a node? borrow from sibling if possible else perform fusion.
  - Now consider deleting 23.



- We can easily generalise the techniques of (2, 4)-Tree to multiway search tree where instead of every internal node having at least 2 and at most 4 children to multiway search trees where every internal node have at least d and at most $2d$ children, where d is some constant.
- Such trees are known by the name B-tree and are used in modern filesystems and database implementations.

- AVL Tree and (2, 4)-Tree are just two examples of balanced search trees.
- There are many more examples of such trees.
- The book gives two other examples: red-black tree and Splay tree.

End