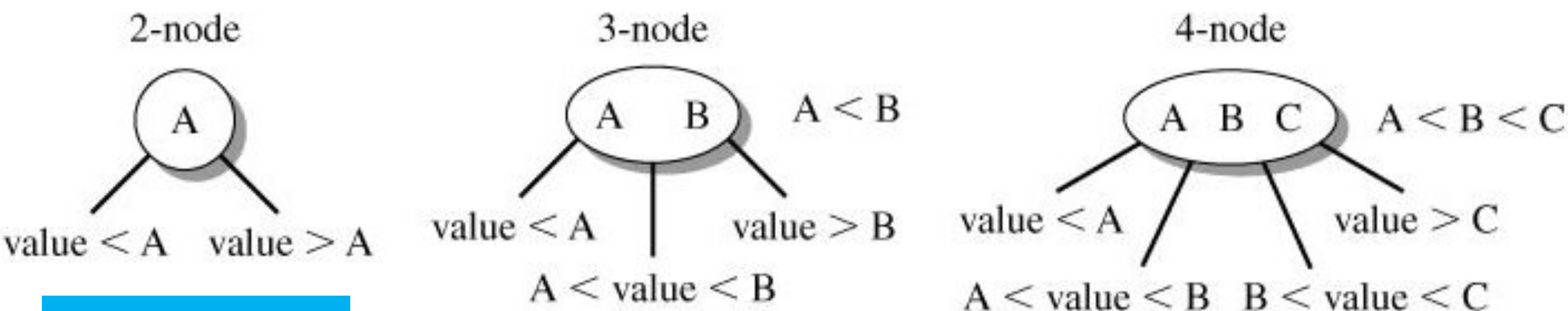
2-3-4 Trees

2-3-4 Trees

- In a 2-3-4 tree:
  - a 2-node has 1 value and a max of 2 children
  - a 3-node has 2 values and a max of 3 children
  - a 4-node has 3 values and a max of 4 children



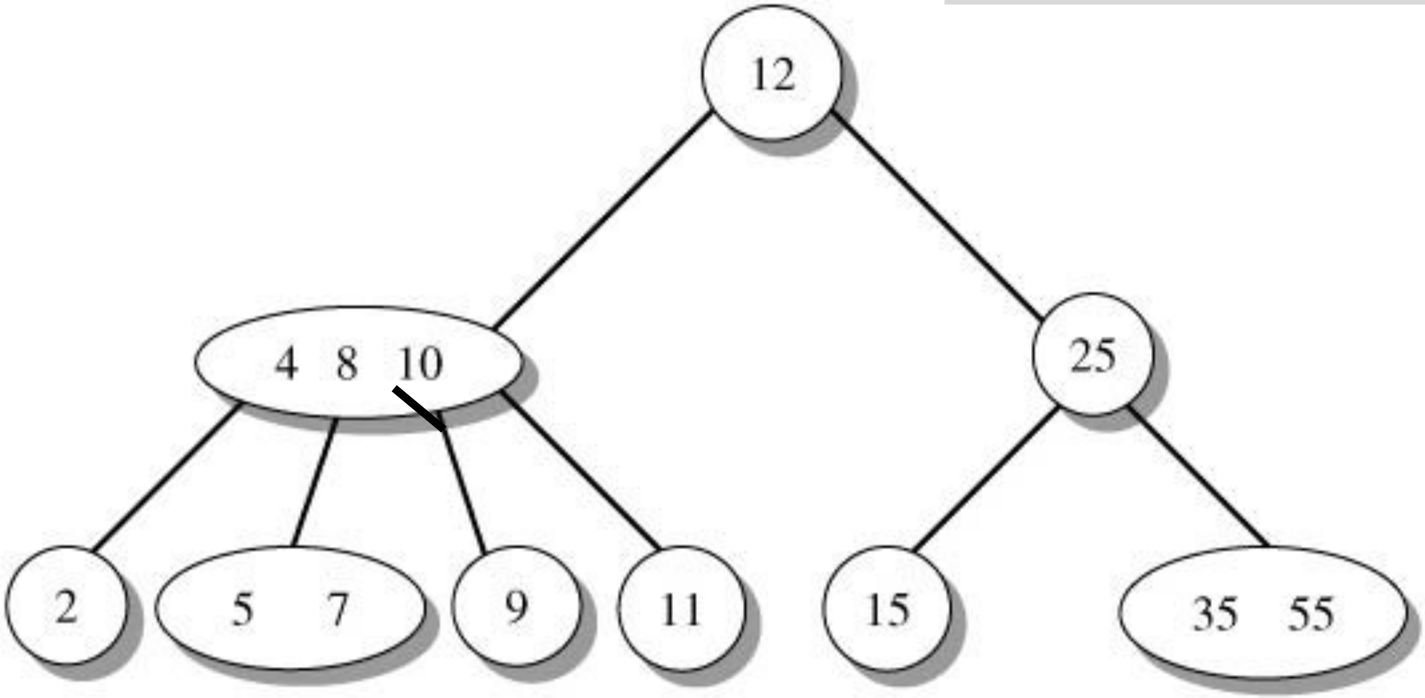
same as a binary tree node

Searching in a 2-3-4 Tree

- To find an item:
  - start at the root and compare the item with <u>all</u> the values in the node;
  - if there's no match, move down to the appropriate subtree;
  - repeat until you find a match or reach an empty subtree

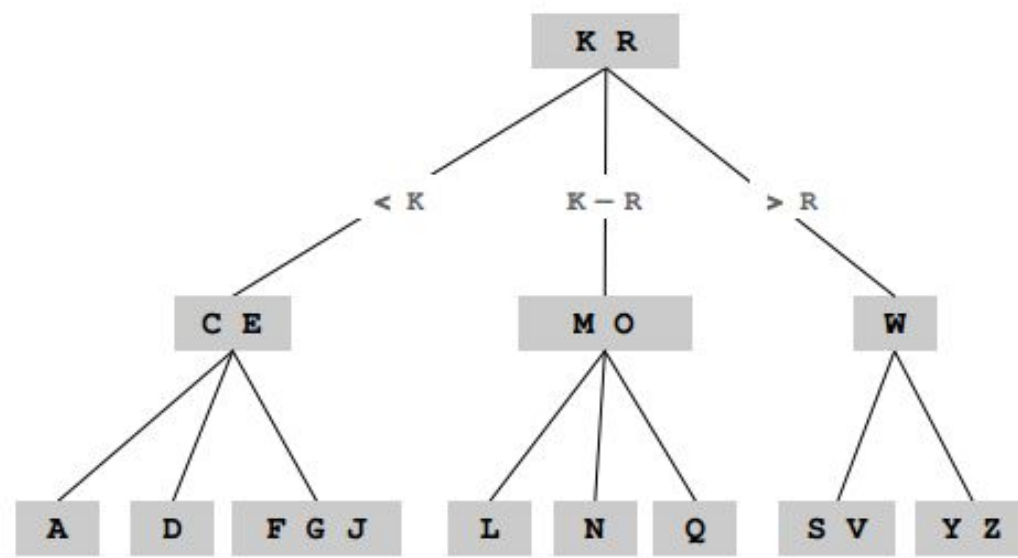
Search Example

Try finding 9 and 30



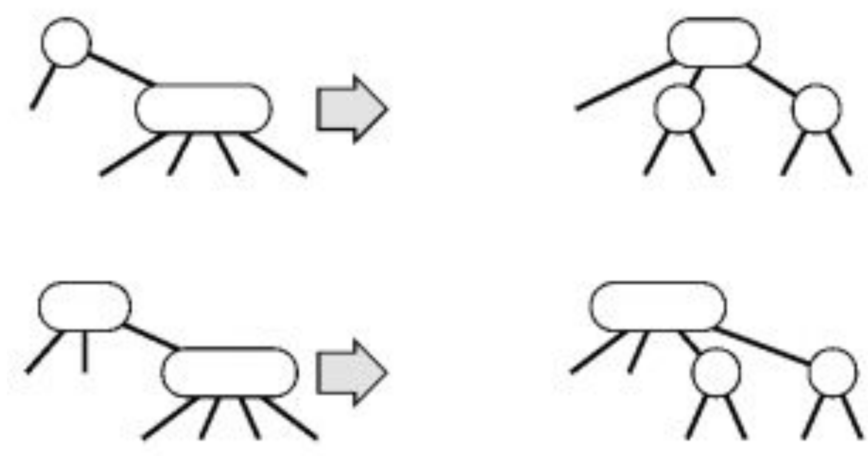
Inserting into a 2-3-4 Tree

- Search to the bottom for an insertion node
  - 2-node at bottom: convert to 3-node
  - 3-node at bottom: convert to 4-node
  - 4-node at bottom: ??



Splitting 4-nodes

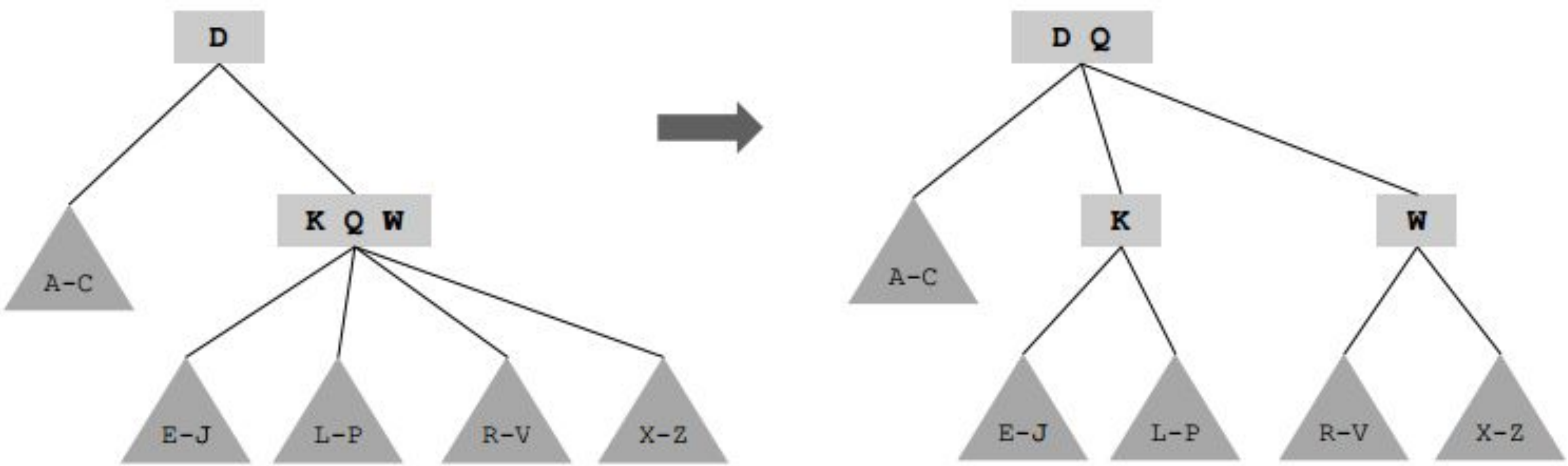
- Transform tree on the way down:
  - ensures last node is not a 4-node
  - local transformation to split a 4-node



Insertion at the bottom is now easy since it's not a 4-node

Example

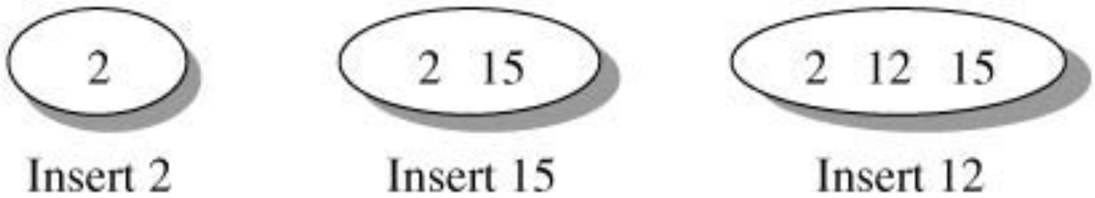
- To split a 4-node. move middle value up.

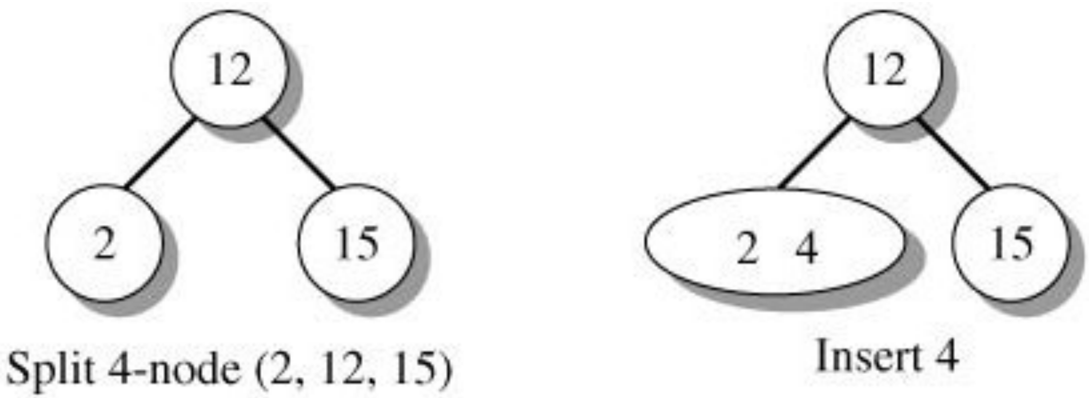


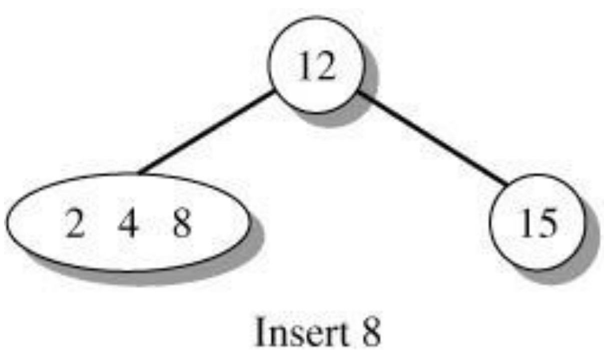
Building

This 4-node will be split during the next insertion.

insert 4





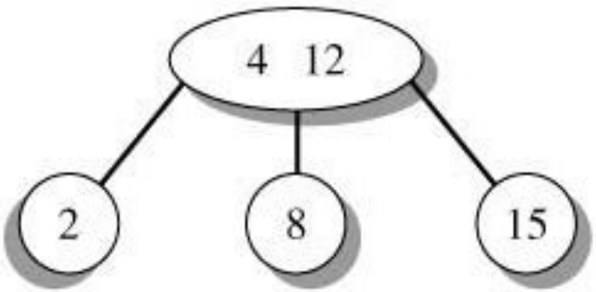


insert 10

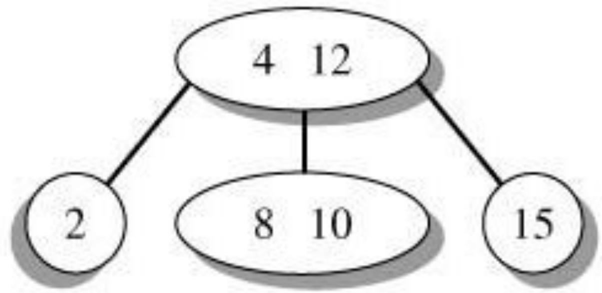
This 4-node will be split during the next insertion.

continued

insert 10

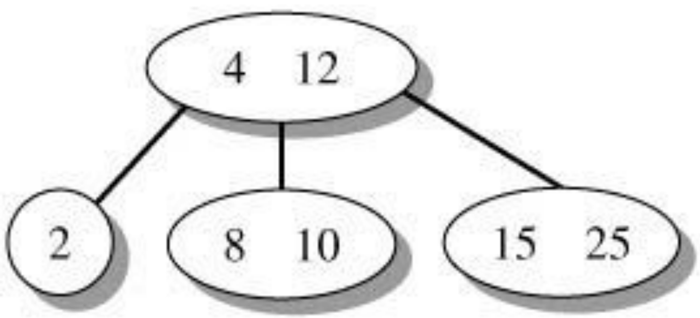


Split 4-node (2, 4, 8)

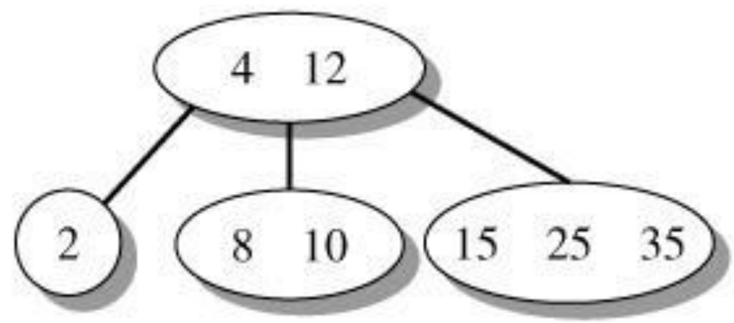


Insert 10

insert 25



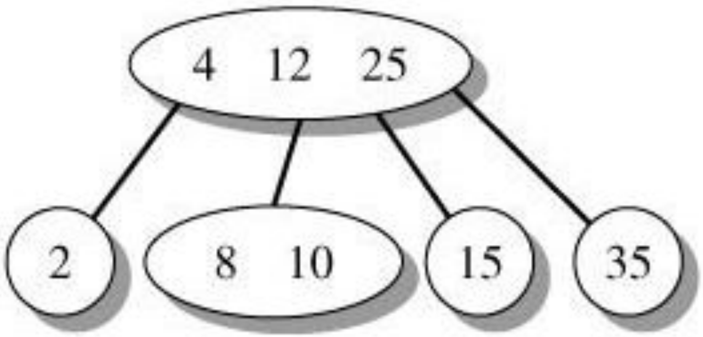
Insert 25



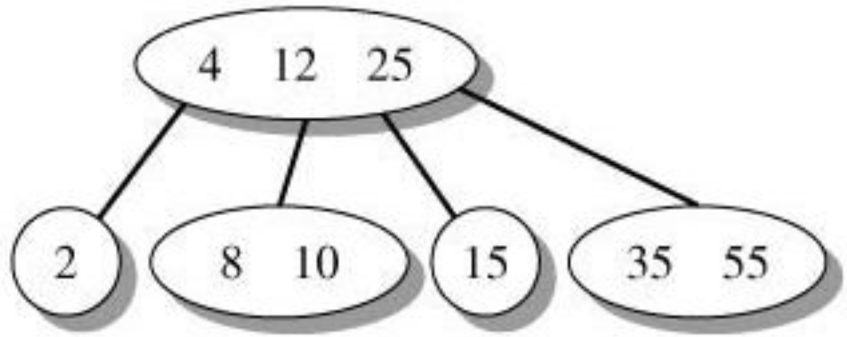
Insert 35

insert 55

insert 55



Split 4-node (15, 25, 35)

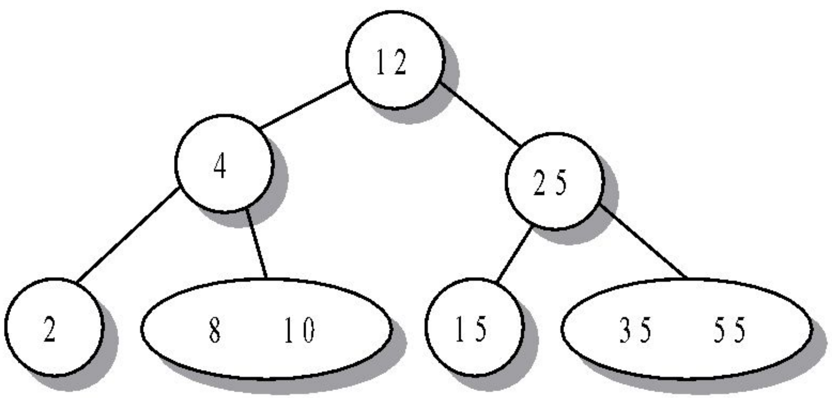


Insert 55

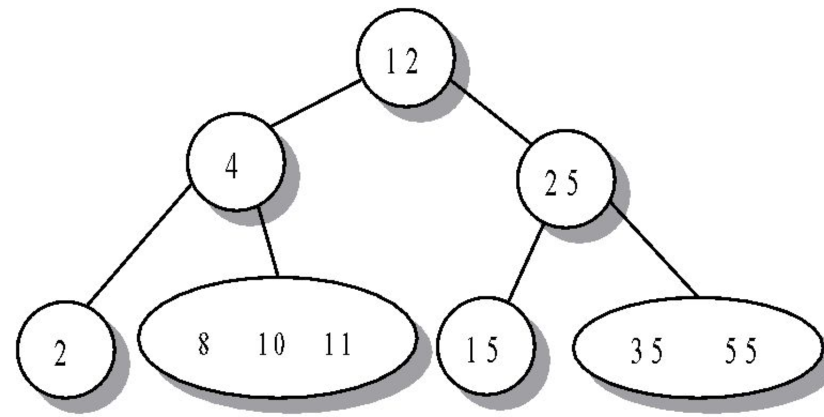
The insertion point is at level 1, so the new 4-node at level 0 is not split during this insertion.

continued

insert 11

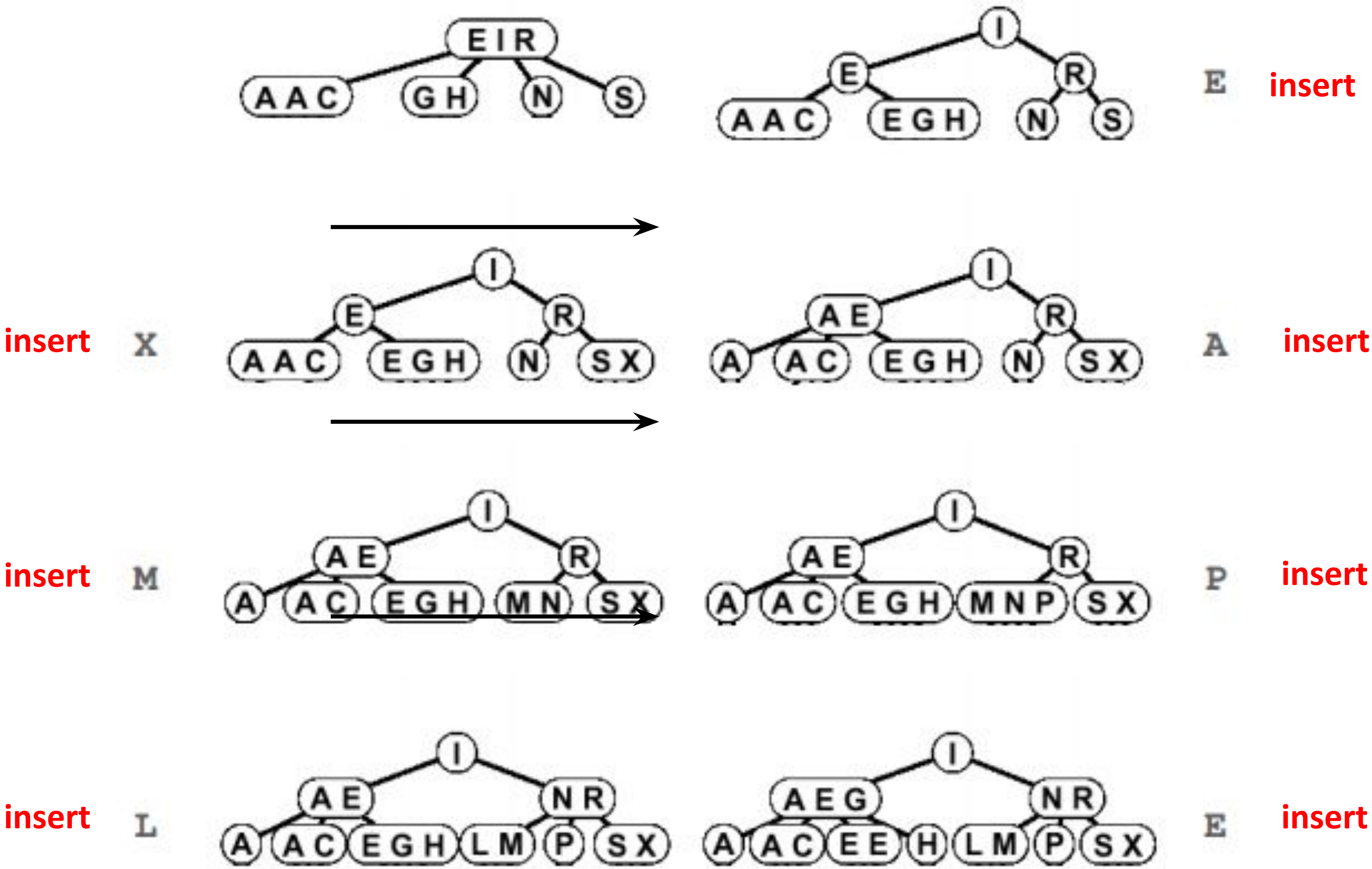


Split 4-node (4, 12, 25)



Insert 11

Another Example



Efficiency of 2-3-4 Trees

- Searching for an item in a 2-3-4 tree with n elements:
  - the max number of nodes visited during the search is $\text{int}(\log_2 n) + 1$

- Inserting an element into a 2-3-4 tree:
  - requires splitting no more than $\text{int}(\log_2 n) + 1$ 4-nodes
    - normally requires far fewer splits

Drawbacks of 2-3-4 Trees

- Since any node may become a 4-node, then all nodes must have space for 3 values and 4 links
  - but most nodes are not 4-nodes
  - lots of wasted memory, unless impl. is fancier
- Complex nodes and links
  - slower to process than binary search trees

R-B Trees

Three Properties of a Red-Black Tree

that must always be true for the tree to be red-black

- 1. The root must always be BLACK (white in our pictures)

- 2. A RED parent never has a RED child
  - in other words: there are never two successive RED nodes in a path

continued

- 3. Every path from a node to an null leaf (node) contains the same number of BLACK nodes
  - called the *black height*

- We can use black height to measure the balance of a red-black tree.

Example in board

- Insertion

  10, 8, 6, 7, 20, 9, 5, 15, 3, 2

Deletion on board