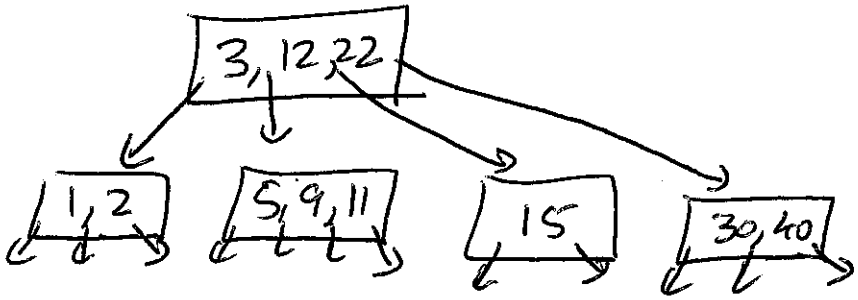
2-4 Trees

Source: Data Structures and Algorithms in Java
Goodrich + Tamassia (9.4)

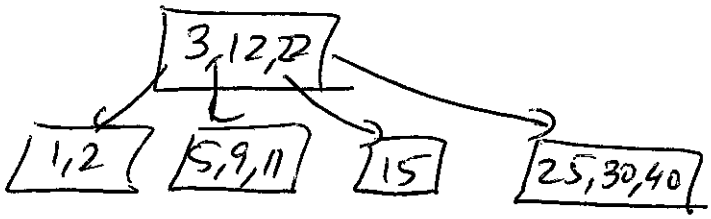
General Structure: B-Tree
2-4 Tree is a specific B-Tree.

Balanced Binary Search Tree

⓪ Search Tree Invariant
① Each leaf node is at the same depth.
② Every internal node has 2-4 children.
(each node stores 1 to 3 values)

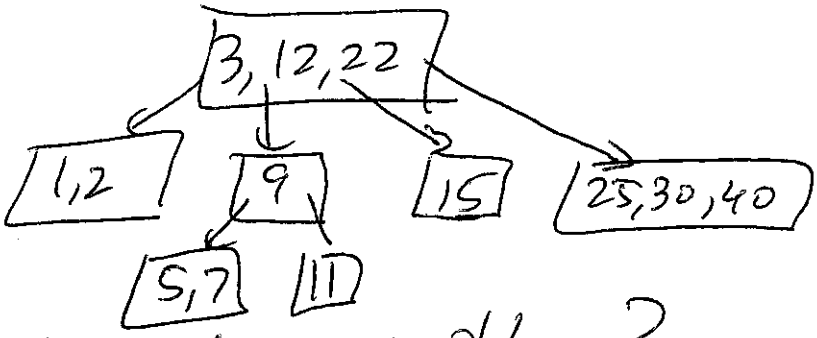


Insertion - ~~easy~~ insert 25 (Case leaf node < 3 values)



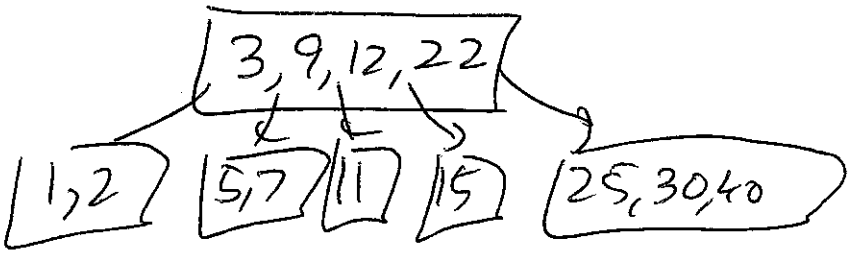
Insertion - hard insert 7 (leaf node has 3 values)

Idea #1



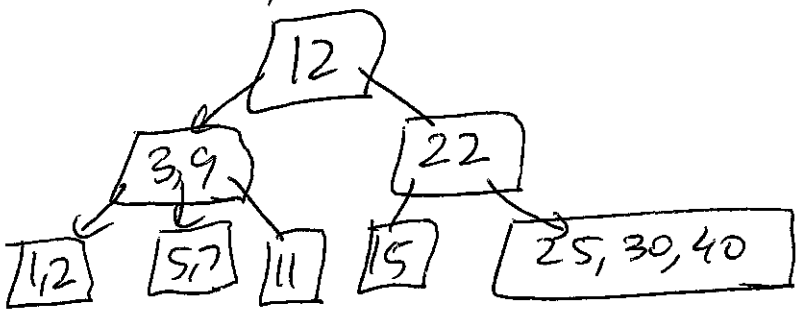
What is the problem?
Leaves on diff levels

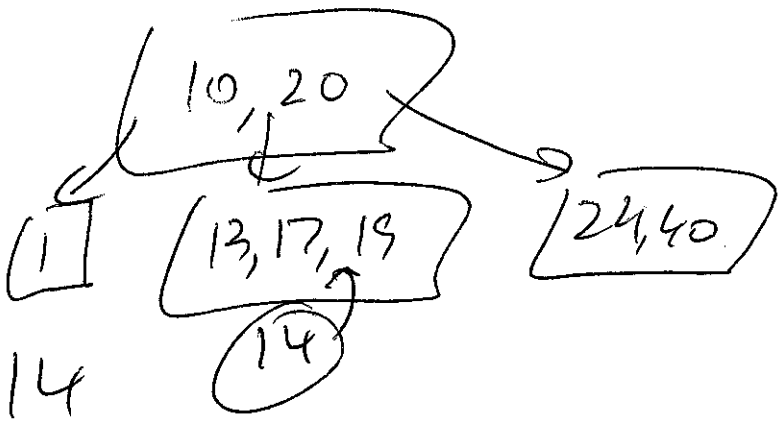
Do this instead:



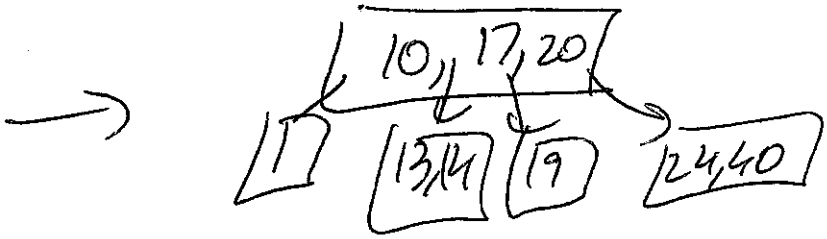
What is the problem?
Root is too full!
Good news: leafs node ~~aren't~~ same level, BST is ok

Note: If top node didn't have 3, this could have completely fixed the issue.





Insert 14



Insertion is $O(\lg n)$

max height is $\lg n$

(If there are $2^h - 1$ nodes, max height is $h-1$.)

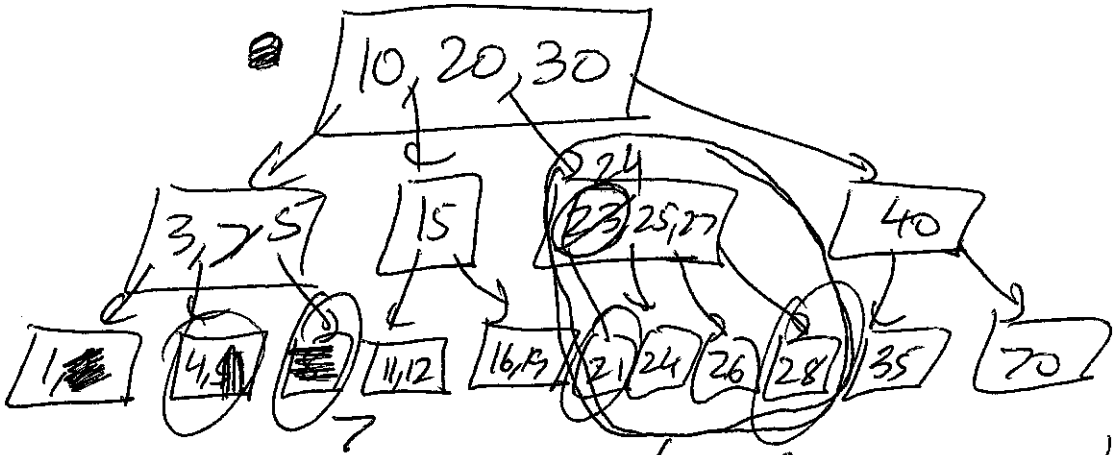
Runtime - $c \lg n$ going down

$c \lg n$ - going back up

$\rightarrow O(\lg n)$.

What I want you to be able to do: Trace through an insertion. (NOT going to code...)

2-4 Tree Deletion



Easy case: Delete 2 (leaf node multiple values)
↳ Just get rid of it.

Next case: leaf node, but only 1 value
Delete 8 (neighbor has > 1 ~~node~~ value)
↳ Transfer Operation

Next: Non-leaf value ~~has~~ 1 value in node. (equivalently make it a leaf node)