(2, 4) Trees

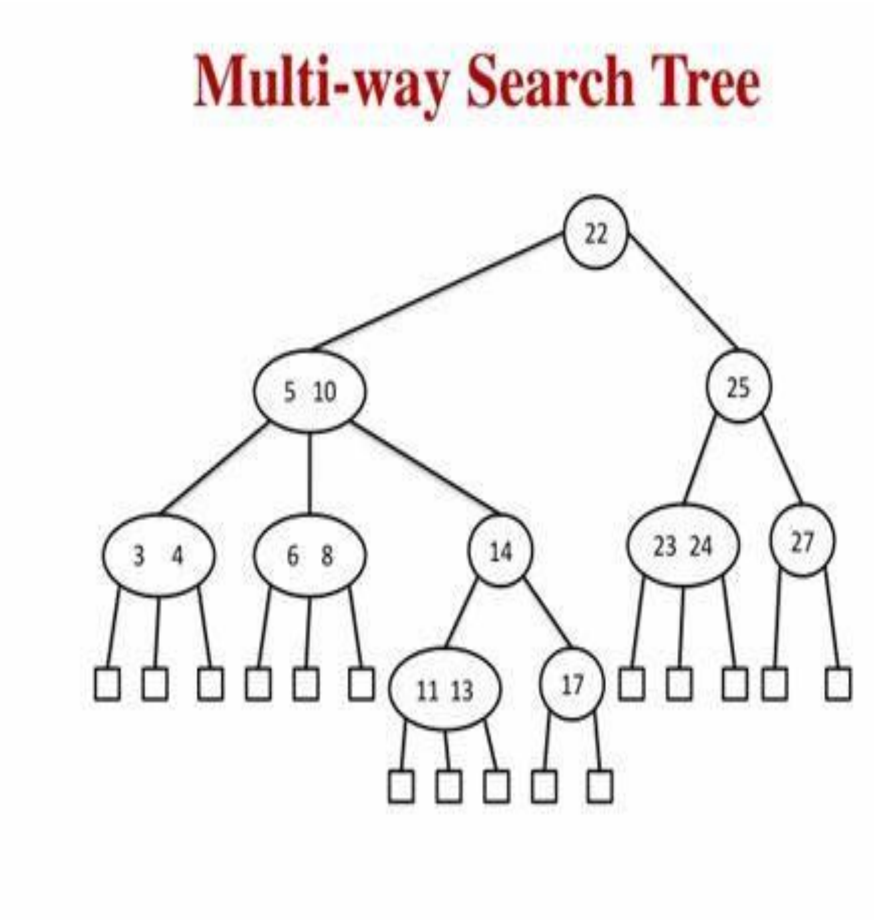
Contents:-

(2, 4) Trees

- They are search trees, but they are not binary search trees
- These 2-4 trees are also called 2-3-4 trees.
- Here 2-3-4 refers to the number of children a node can have a node can have either 2, 3 or 4 children.
- Such trees in which a node can have many children but satisfy a certain kind of search properties are called multi-way search trees.

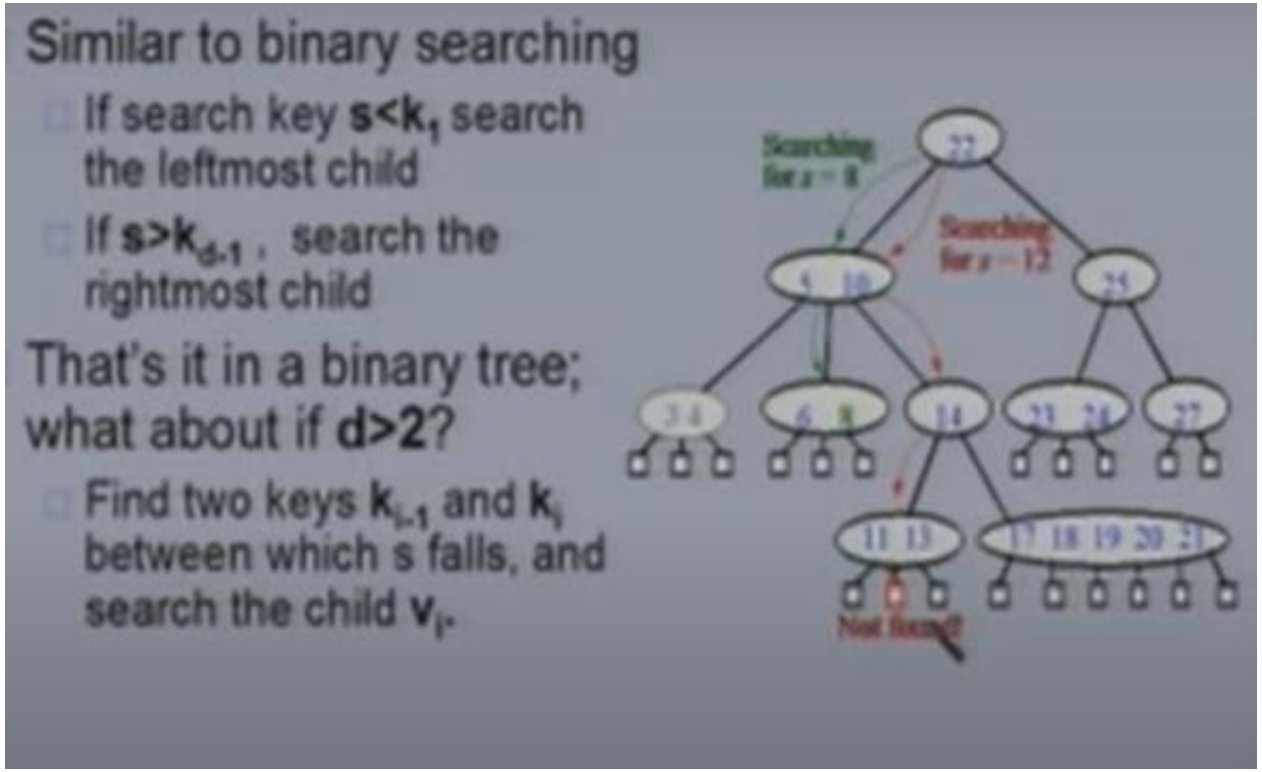
Multi-way search tree

- Each internal node of a multi-way search tree has at least two children. It will have at least two which means it could have more than two children.
- Each node of a tree also stores a collection of items of the form (key, element).
- In the binary search tree there is only one such pair in each node and in a multi-way search tree there could be more than one.
- In particular there could be d-1 such pairs or items, where d is the number of children that particular node has.



Multi-way searching

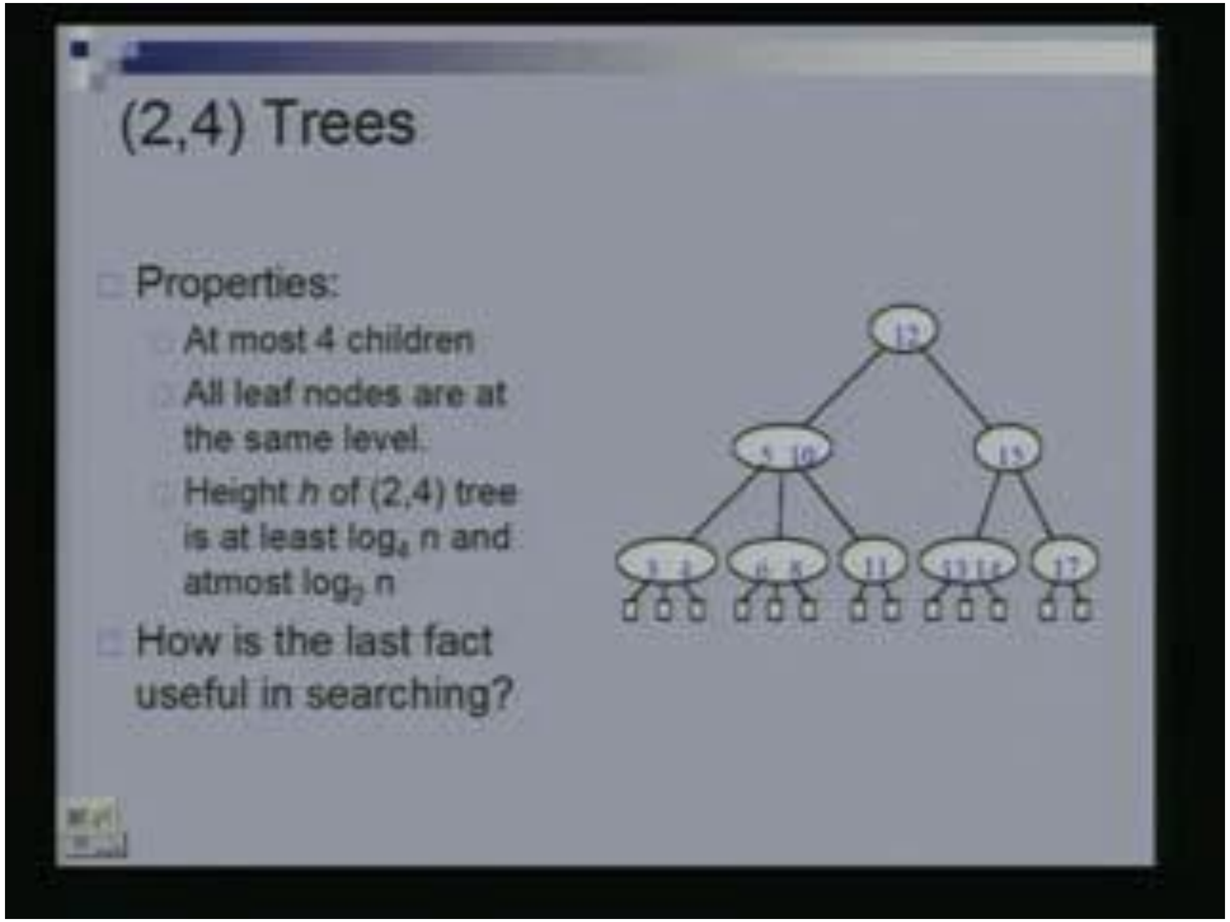
- So, suppose we are searching for 8. You come down here, compare 8 with 22 so 8 is less. So you go here, now you will have to find, so 8 is not less than 5 and 8 is not more than 10. But 8 lies between 5 and 10. So you will follow this and then you will find that 8 is sitting here. So it’s a successful search.



(2, 4) Trees

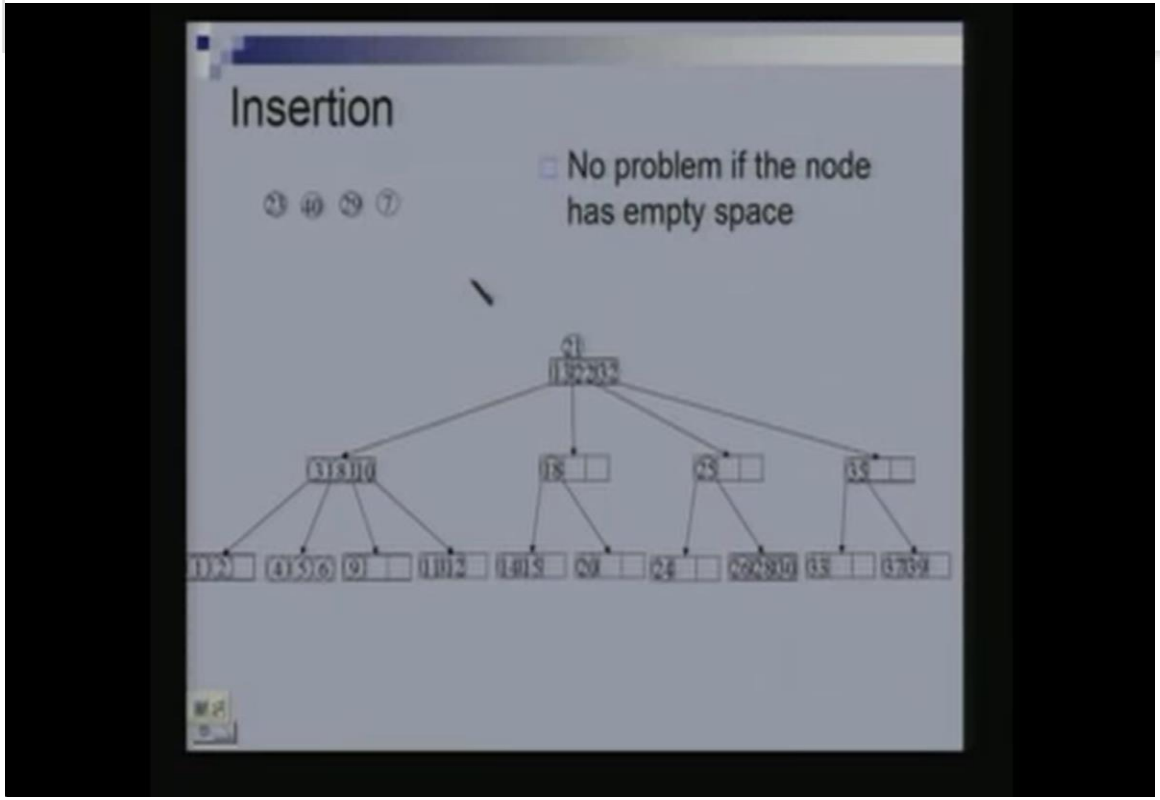
- (2, 4) Tree is same as multiway search tree with 2 additional properties.

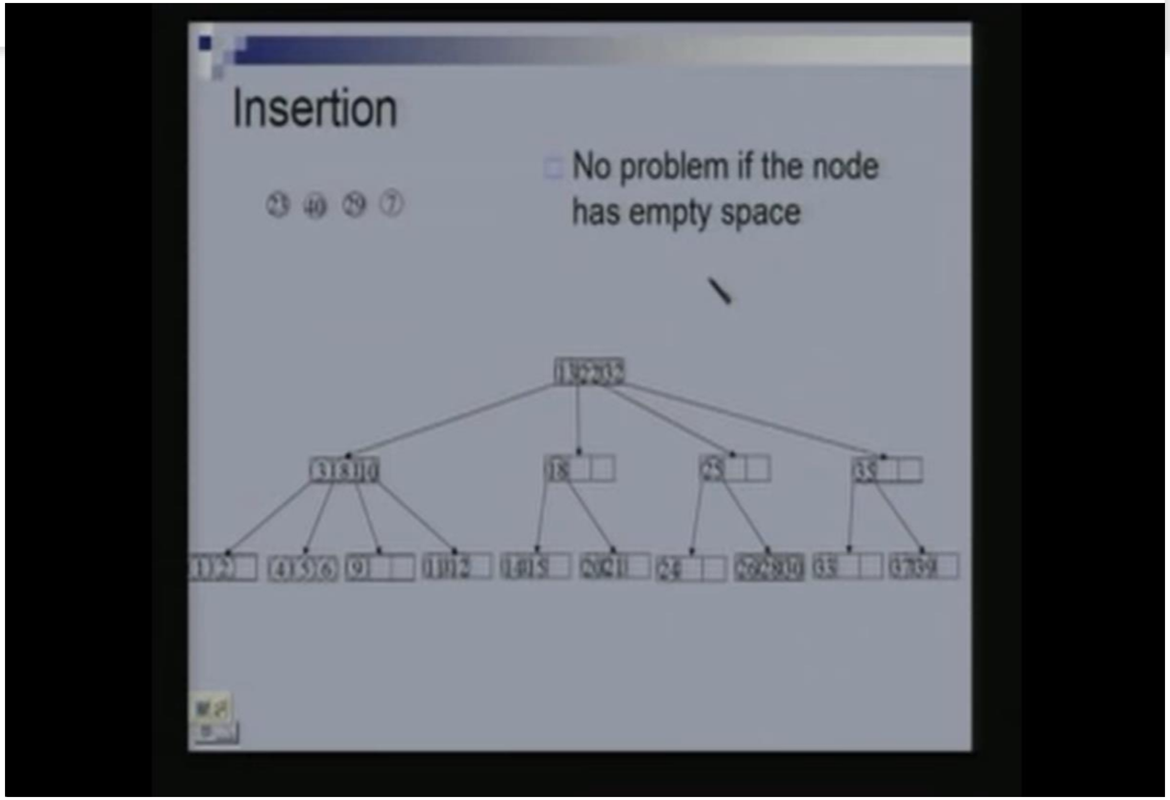
- if we do the same analysis as for complete binary tree you will find that the height of this tree is $\log_4 n$. So height of the 2-4 tree on n nodes always lies between these two quantities.
- So how much time does it take for me to search in a 2-4 tree? Height of the tree.
- So the time is 3 comparisons with in a node, times log n because that is log n is a number of node I would be visiting O(log n)



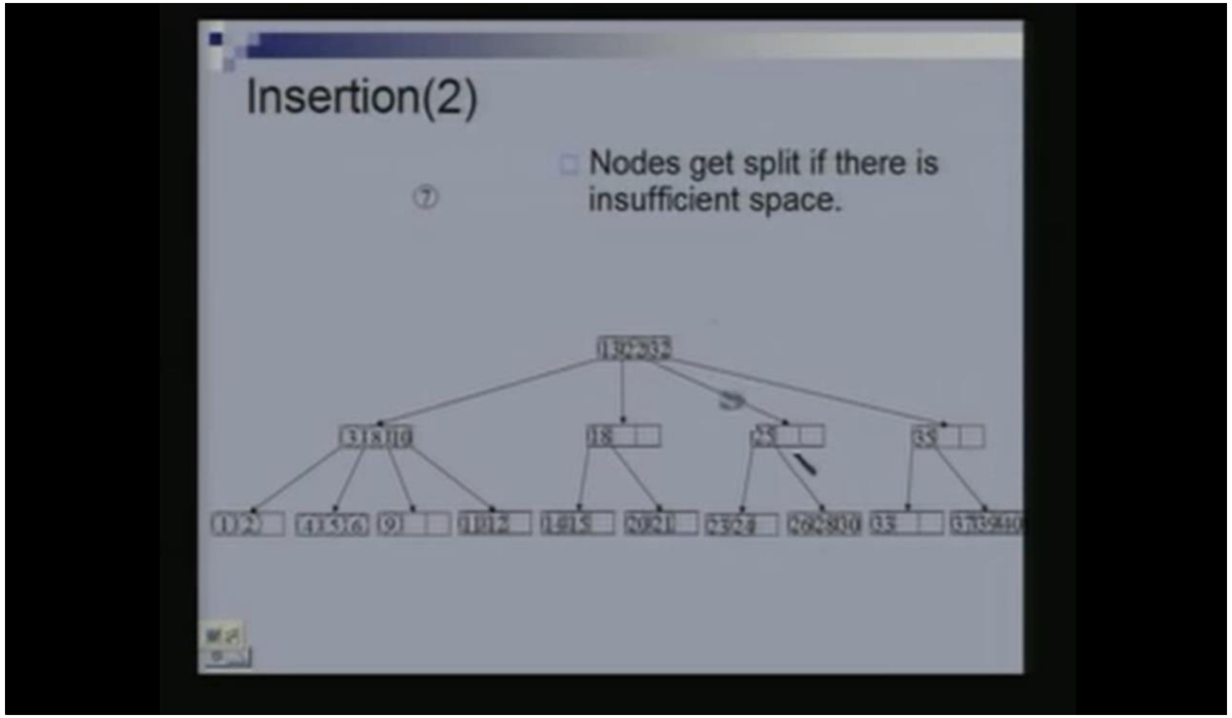
Insertion in a (2,4) tree

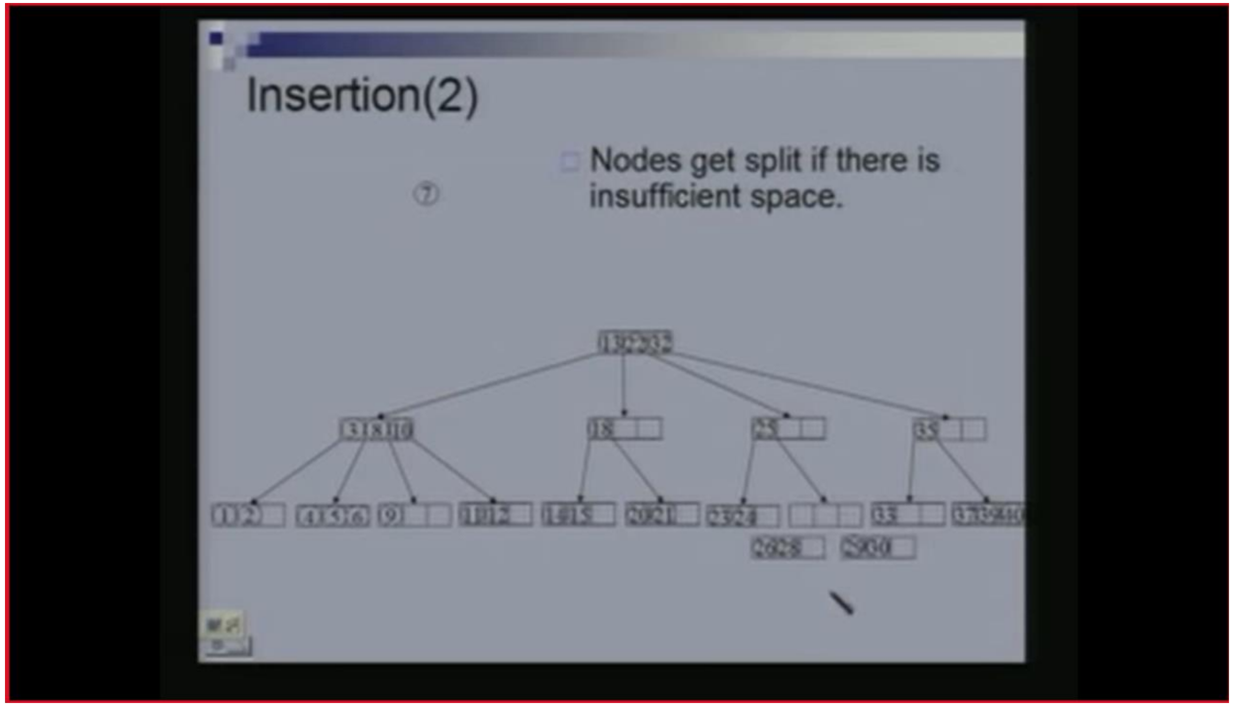
- Case 1:-

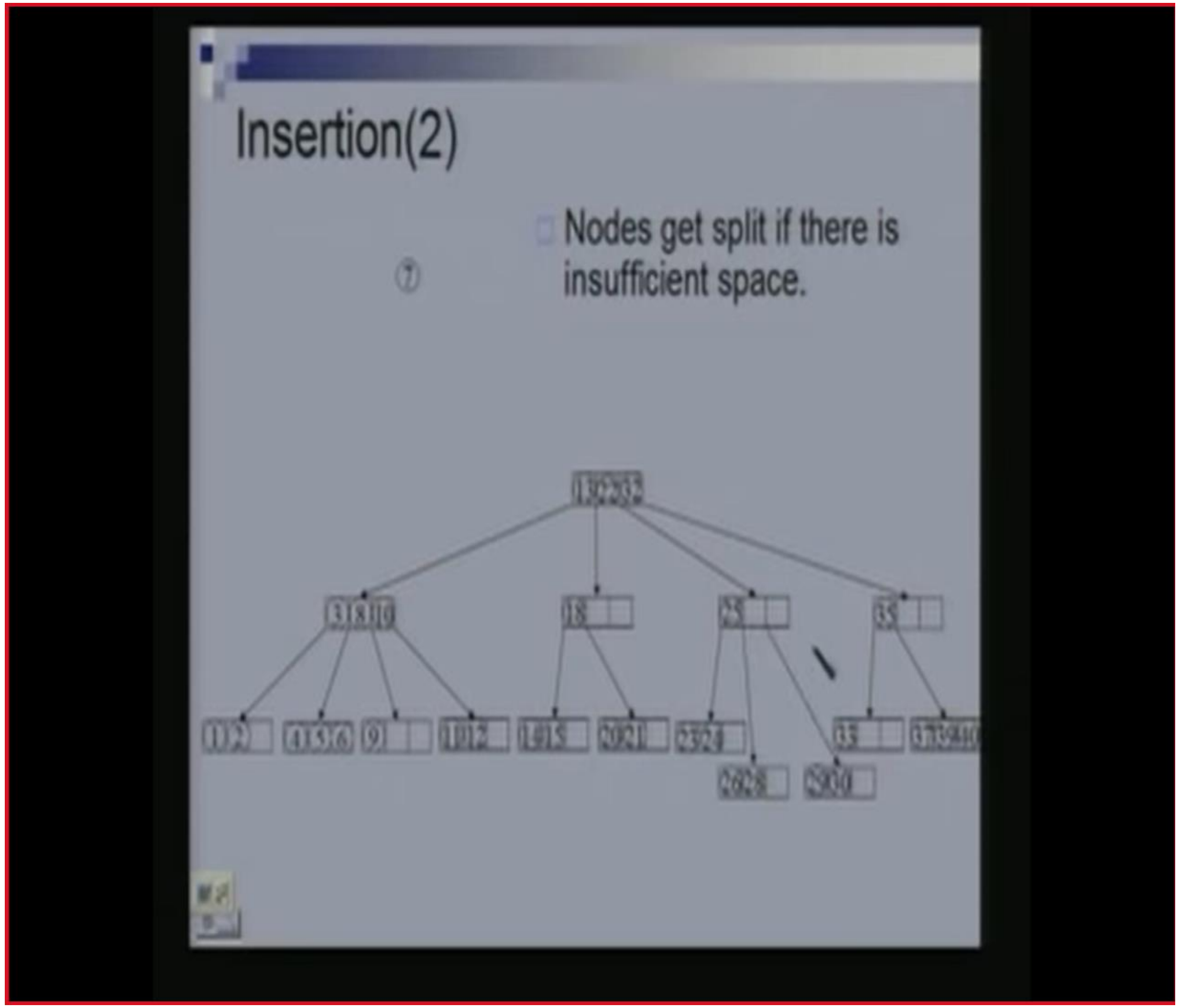


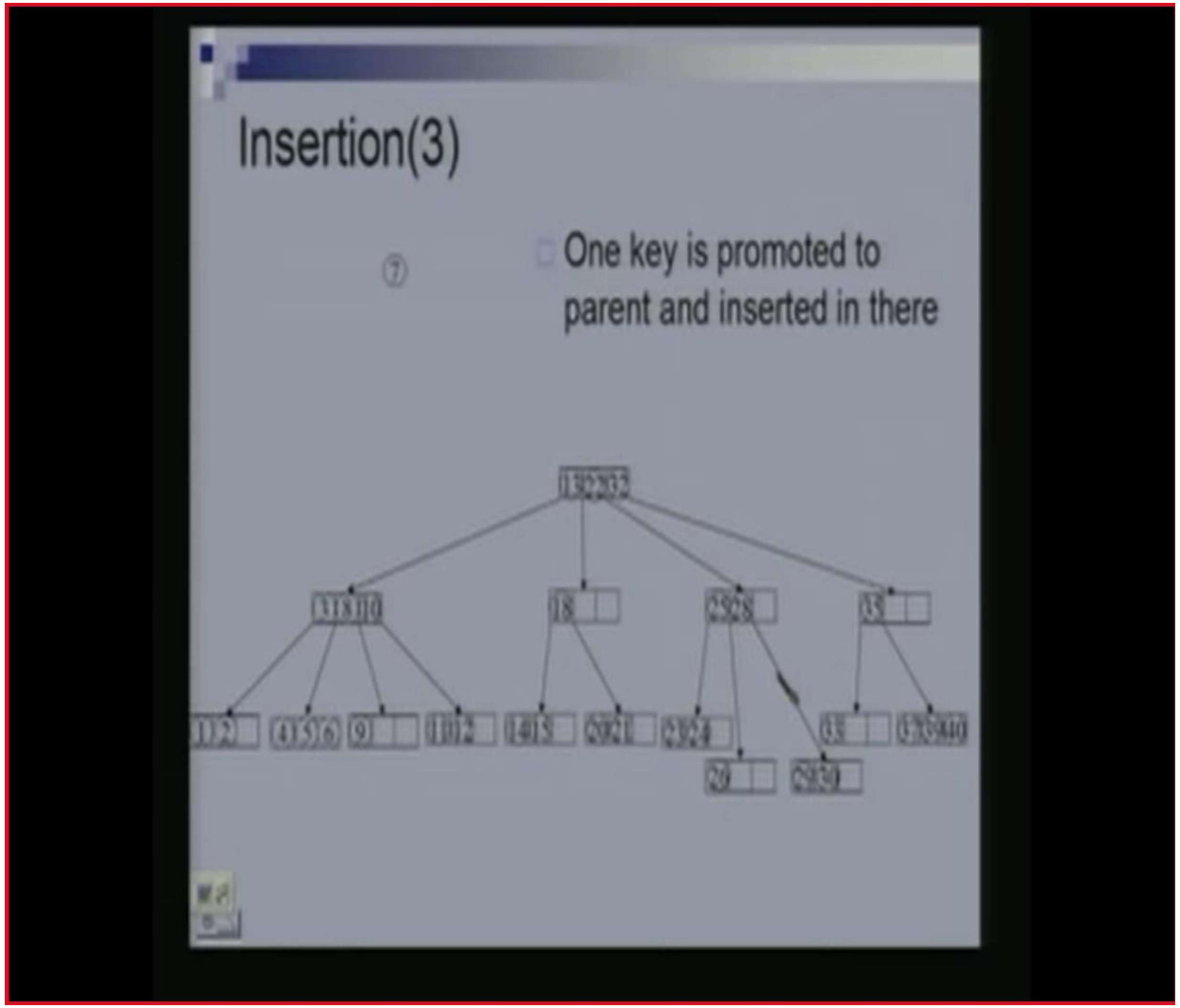


Case 2:-

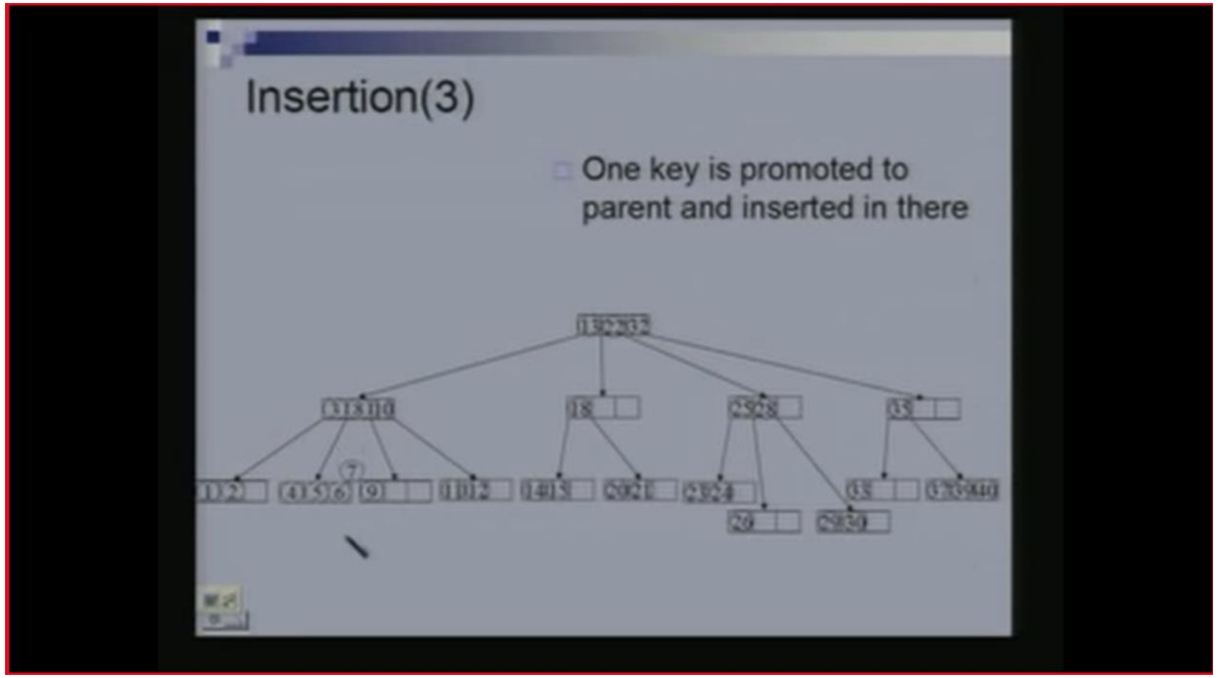

Insertion(2)
Nodes get split if there is insufficient space.

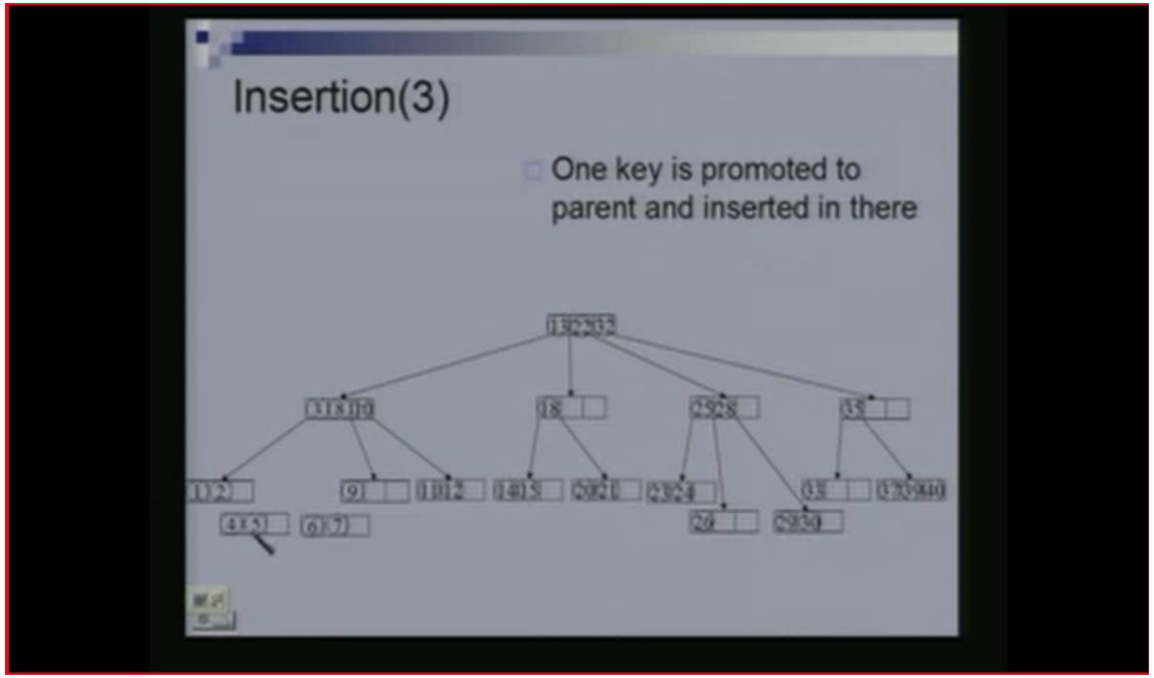

Insertion(2)
Nodes get split if there is insufficient space.


Insertion(2)
Nodes get split if there is insufficient space.

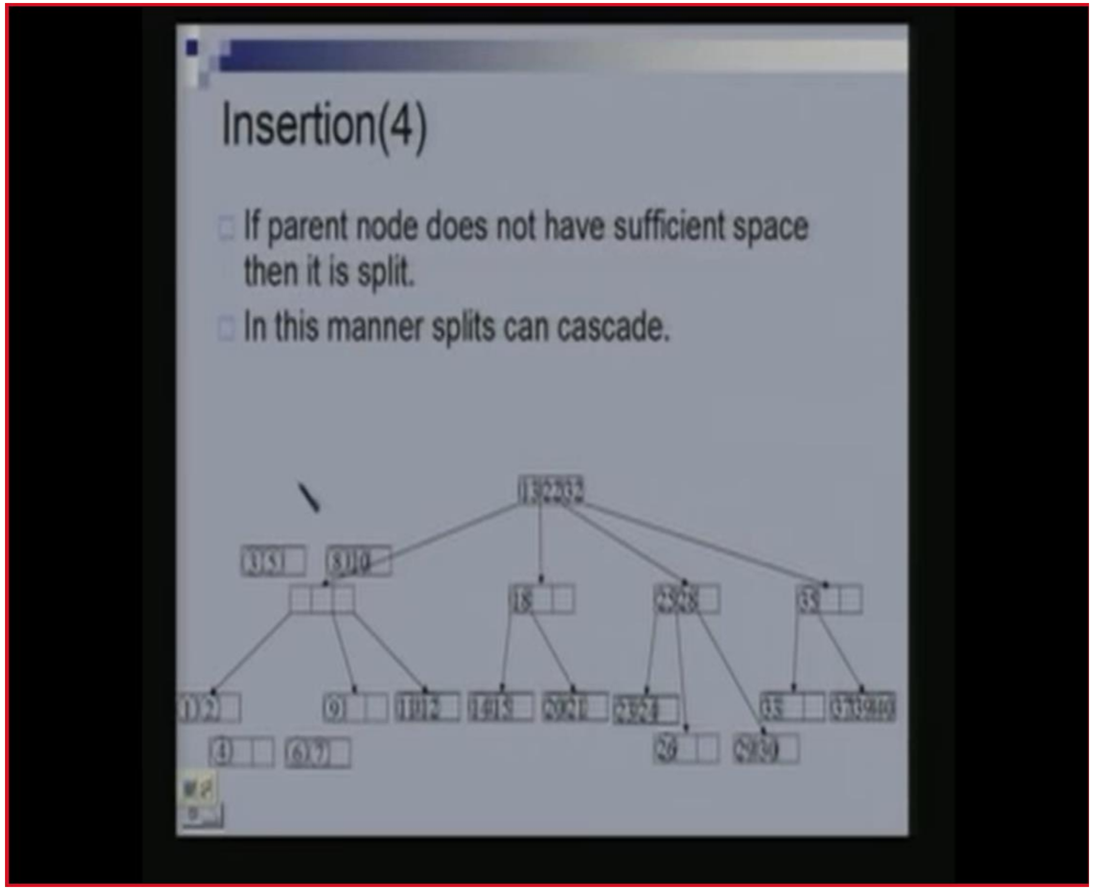

Insertion(3)
One key is promoted to parent and inserted in there

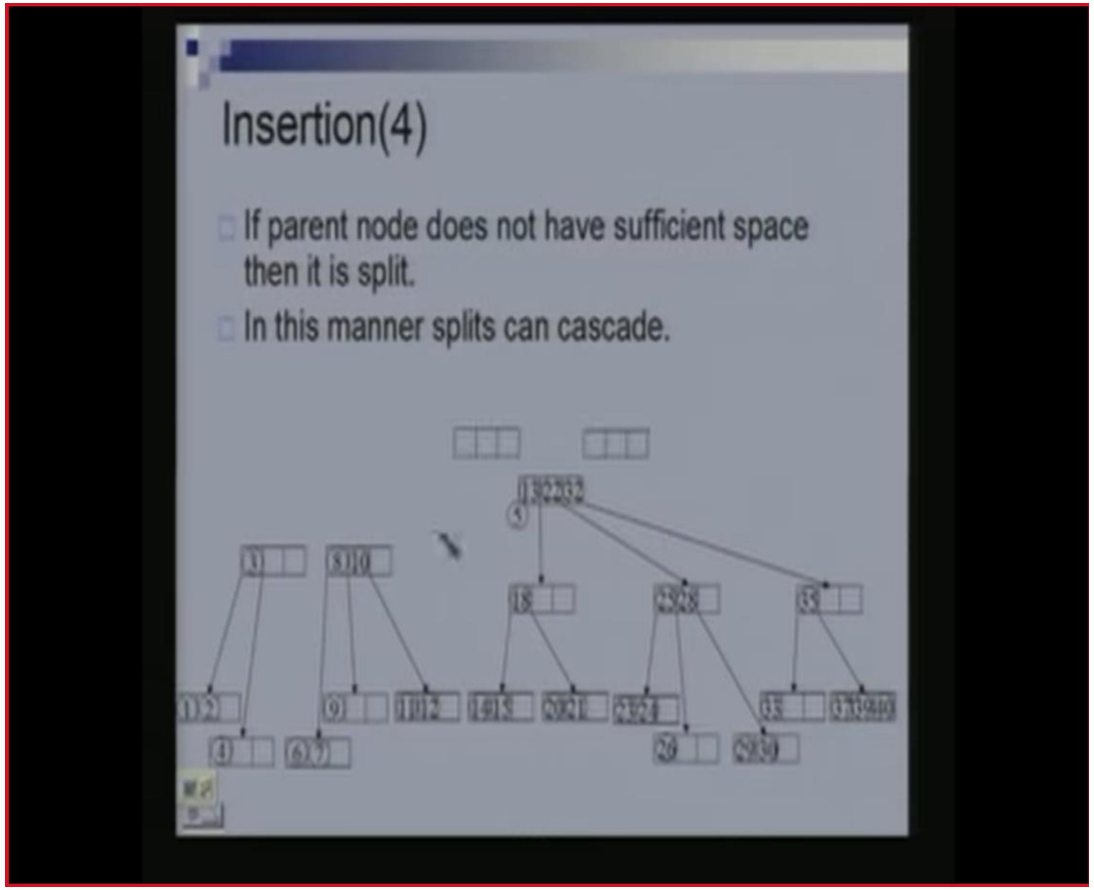
Case 3 the parent is also full



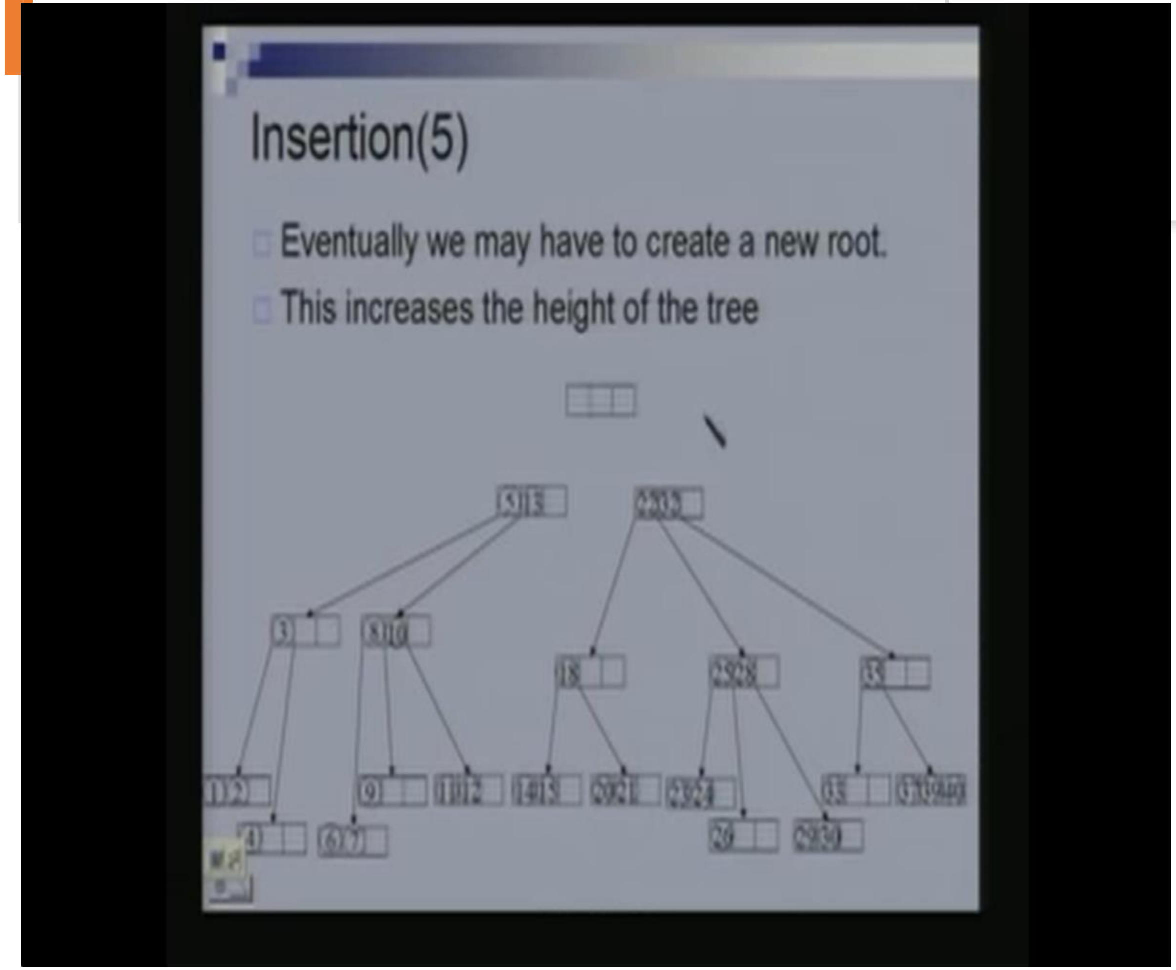


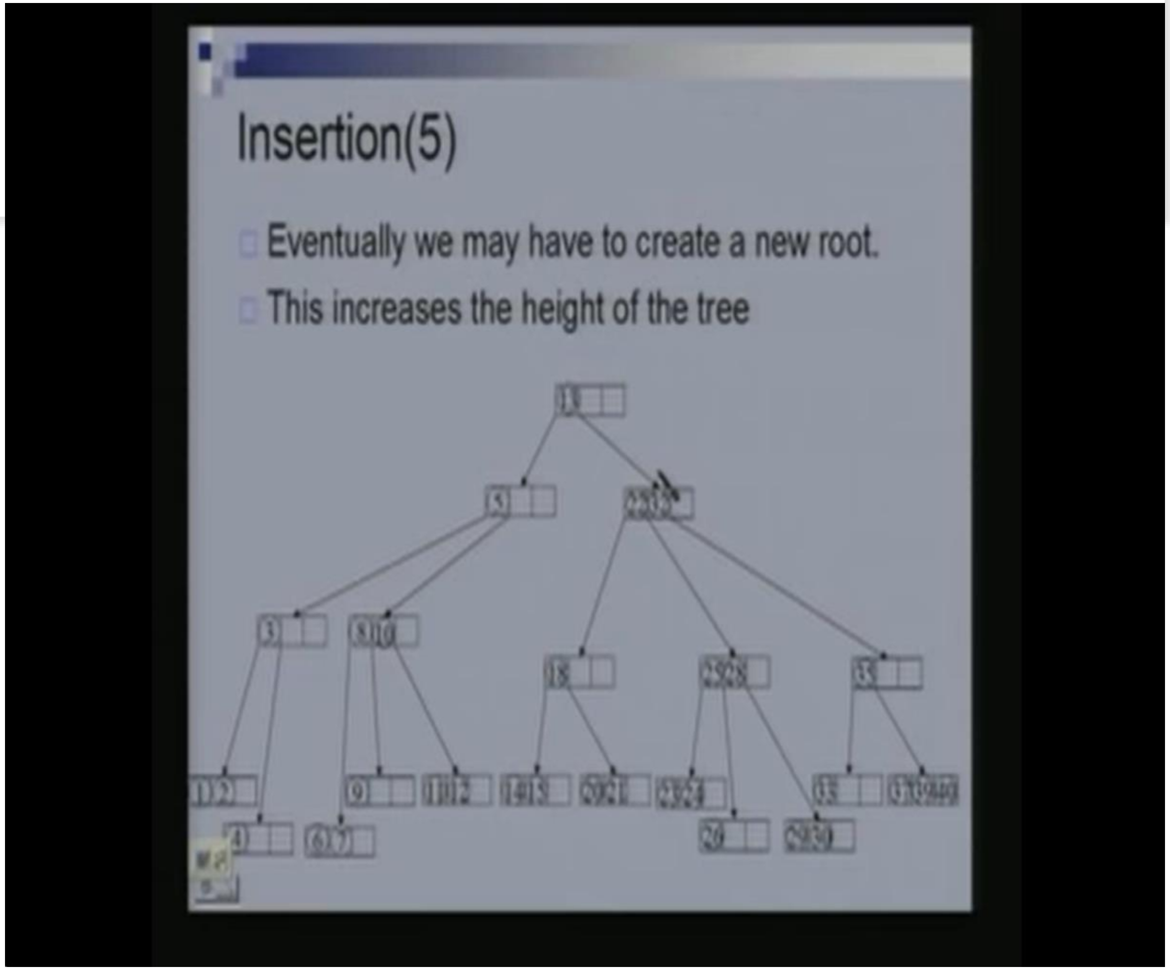
Insertion Continue



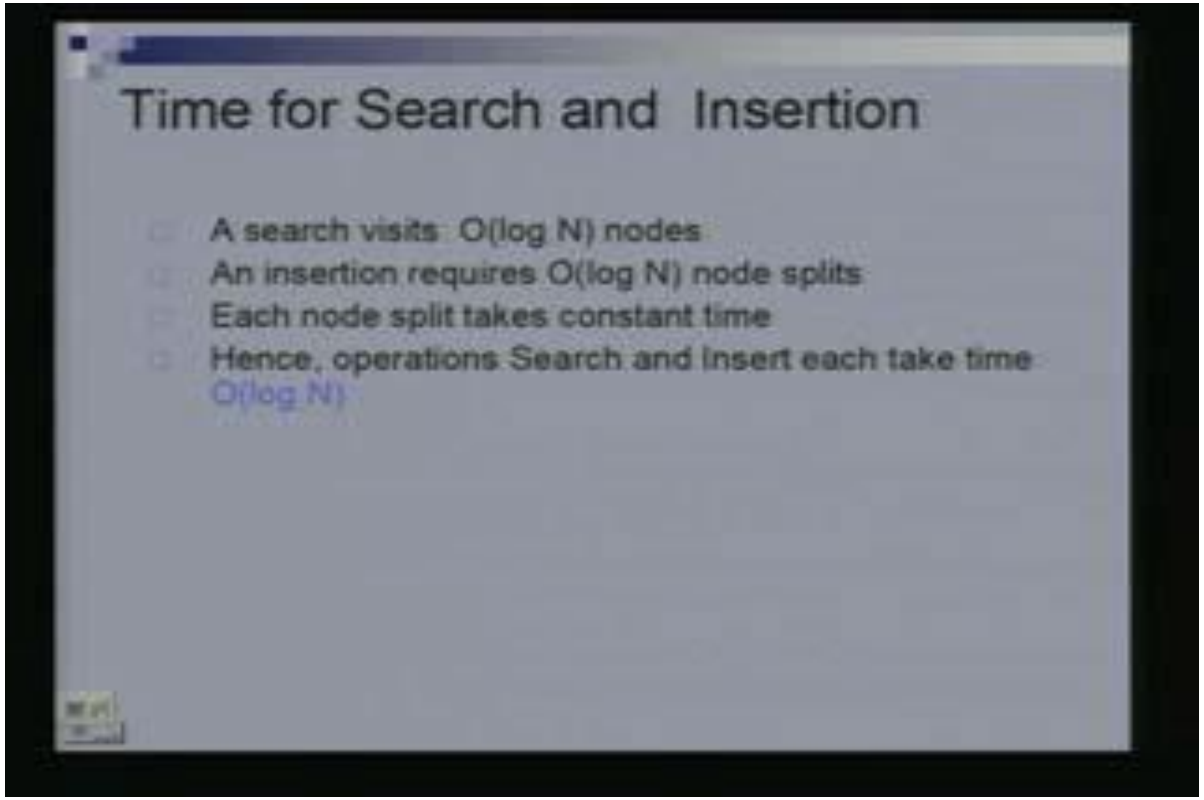


Insertion Continue



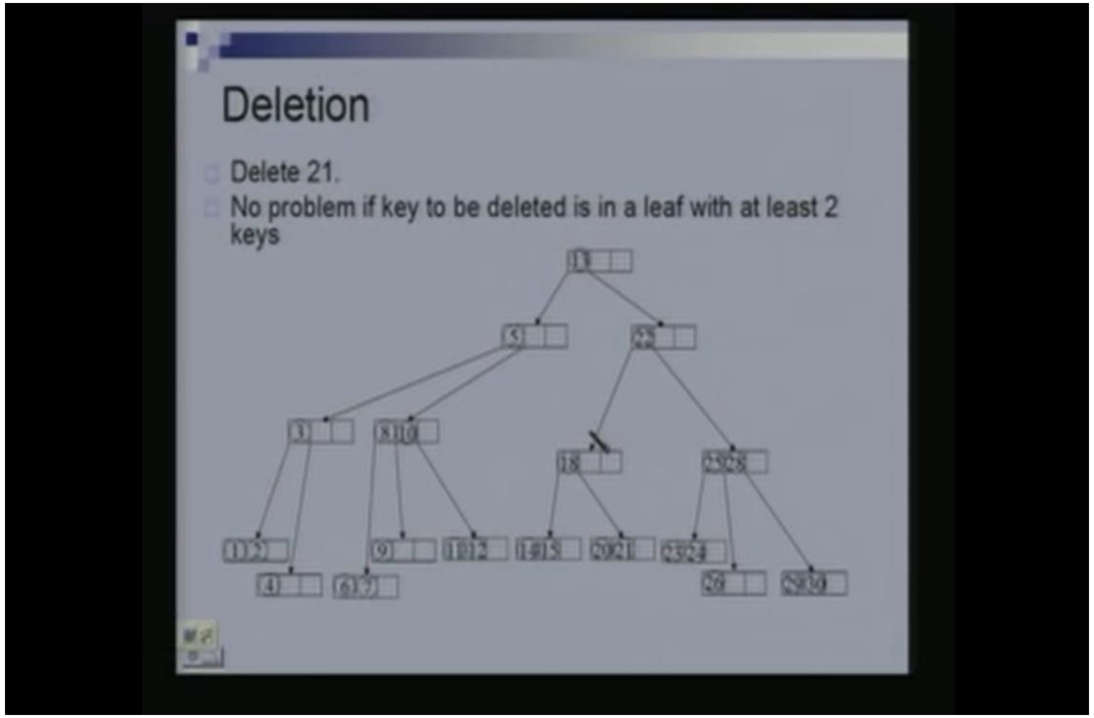


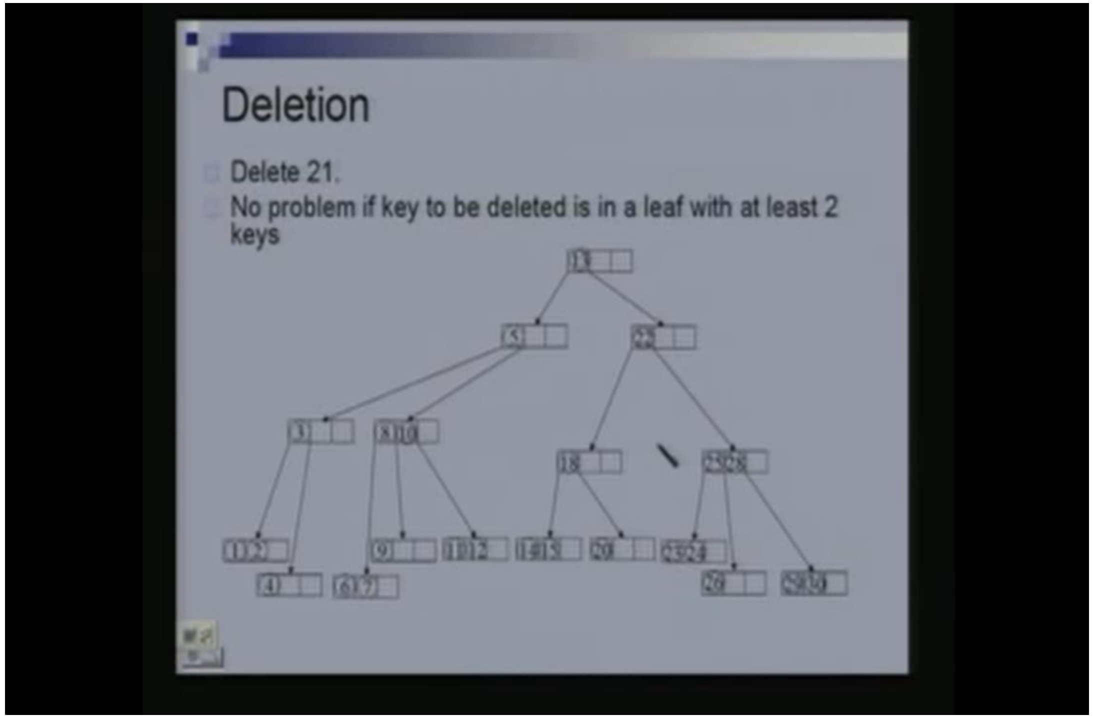
Time complexity for (2,4 tree)



Deletion:-

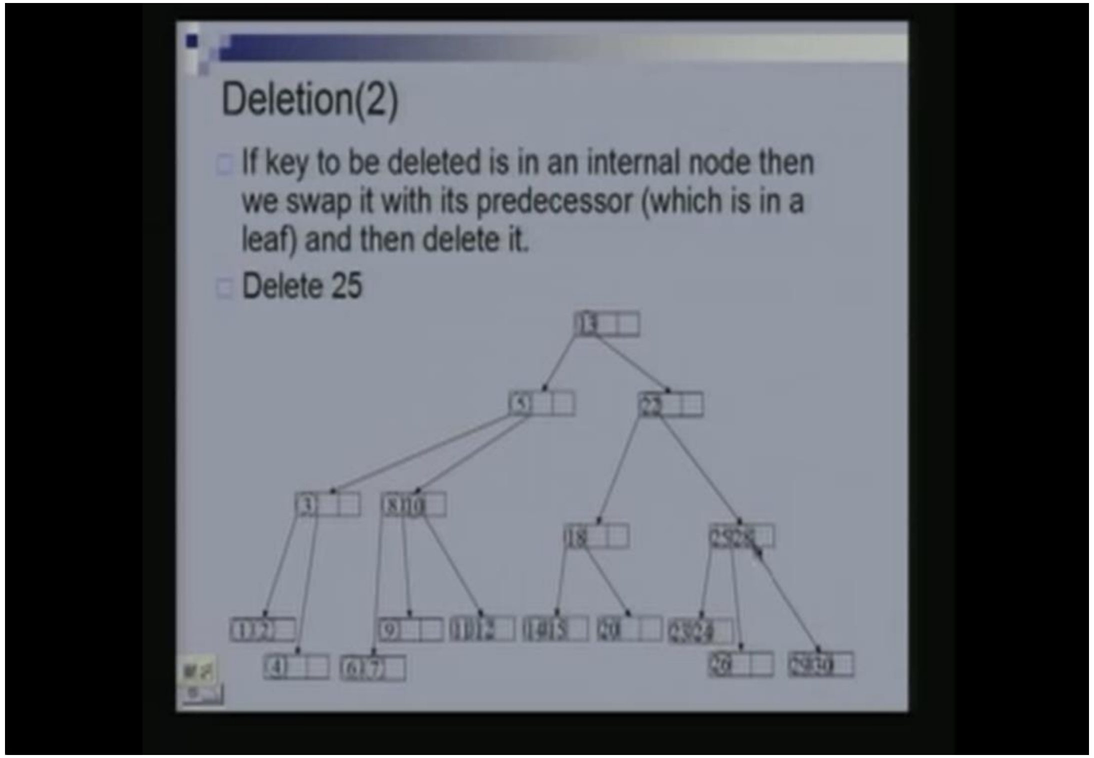
- Case 1 :-
- Leaf node with at least 2 keys.

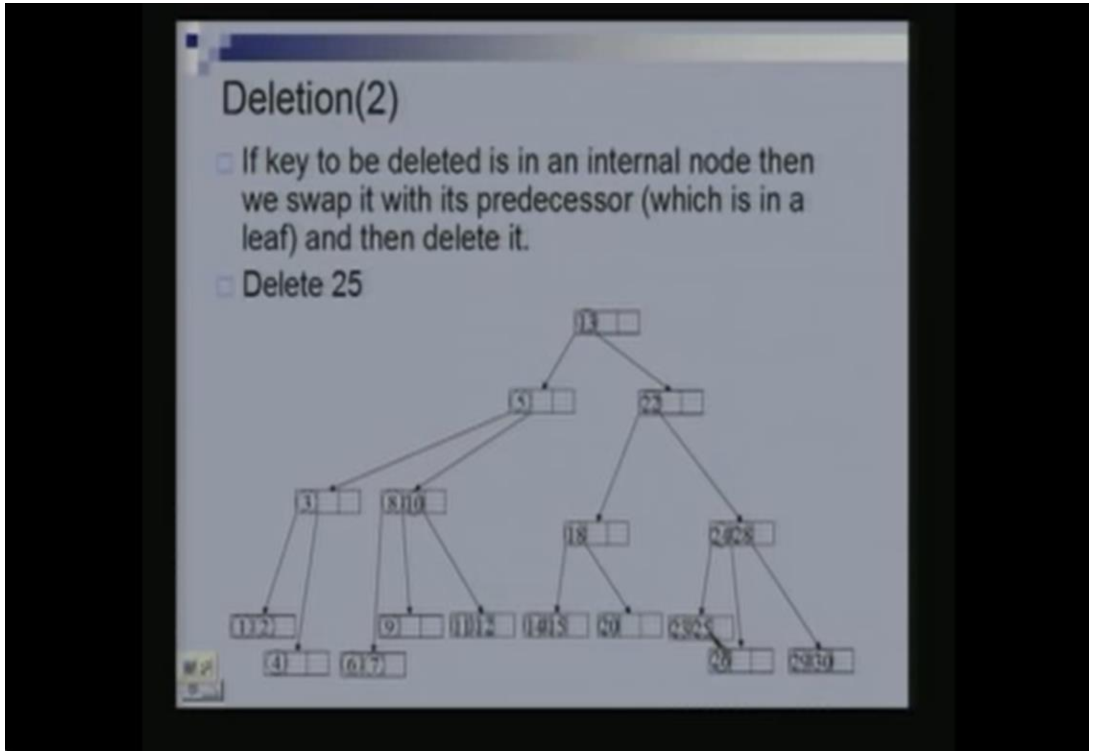




Deletion

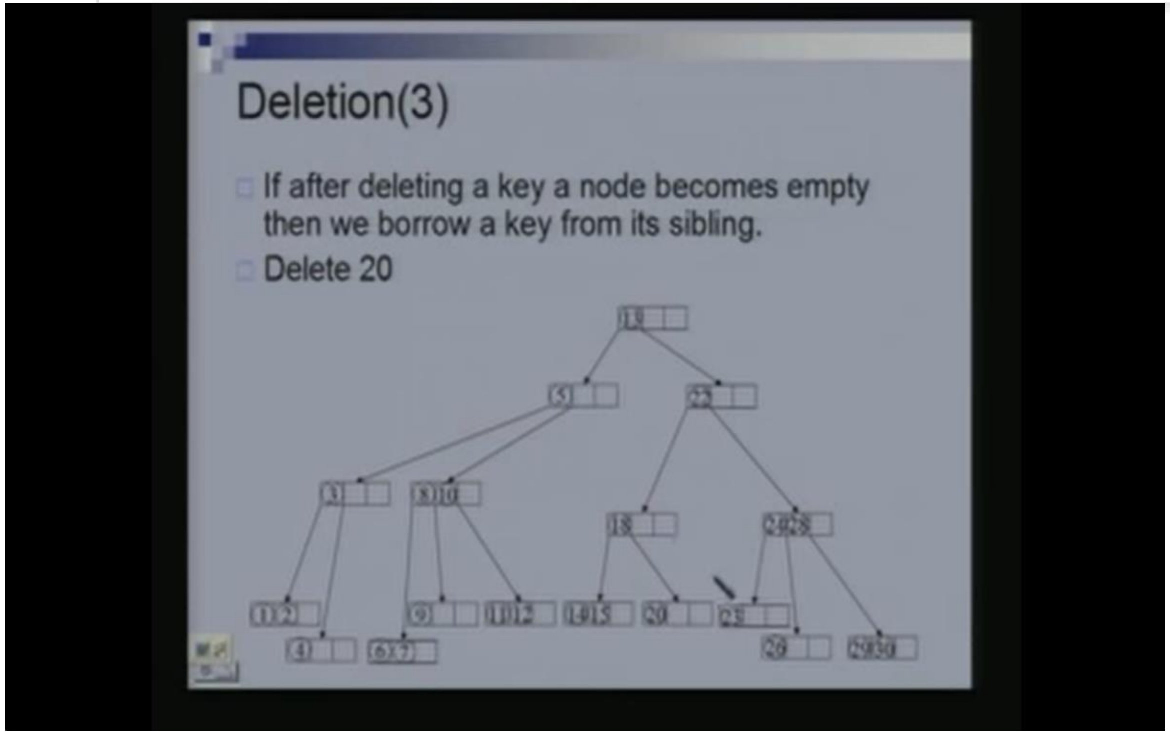
- Case 2:- internal node

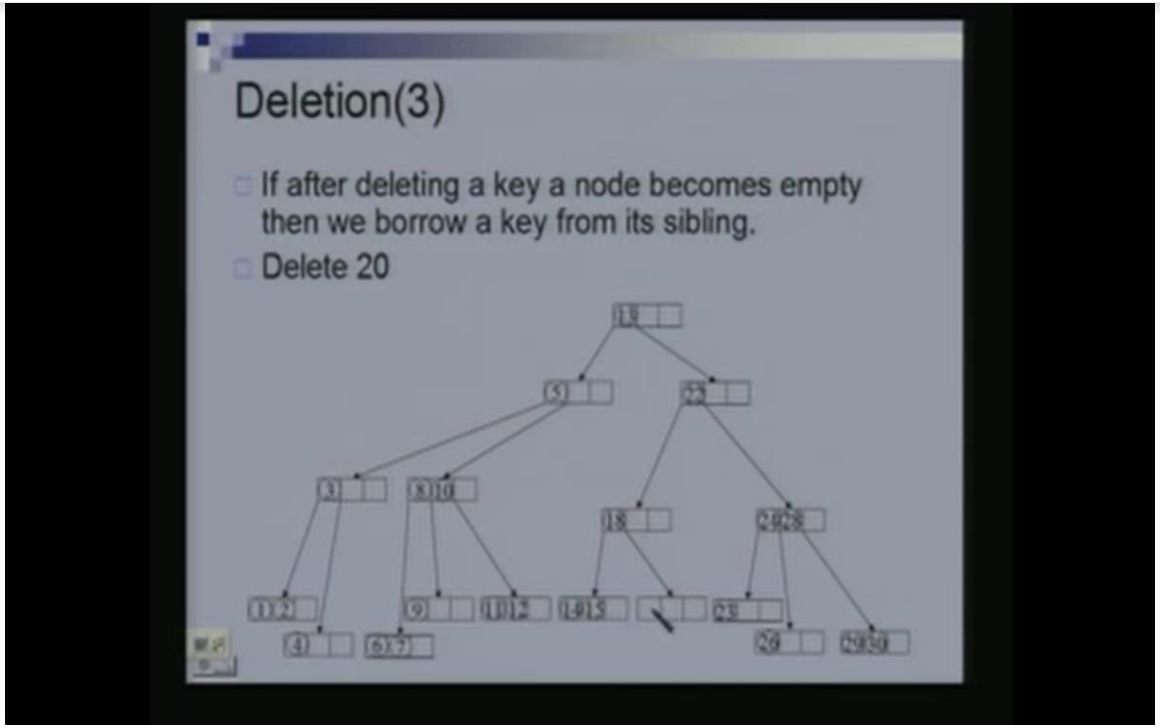




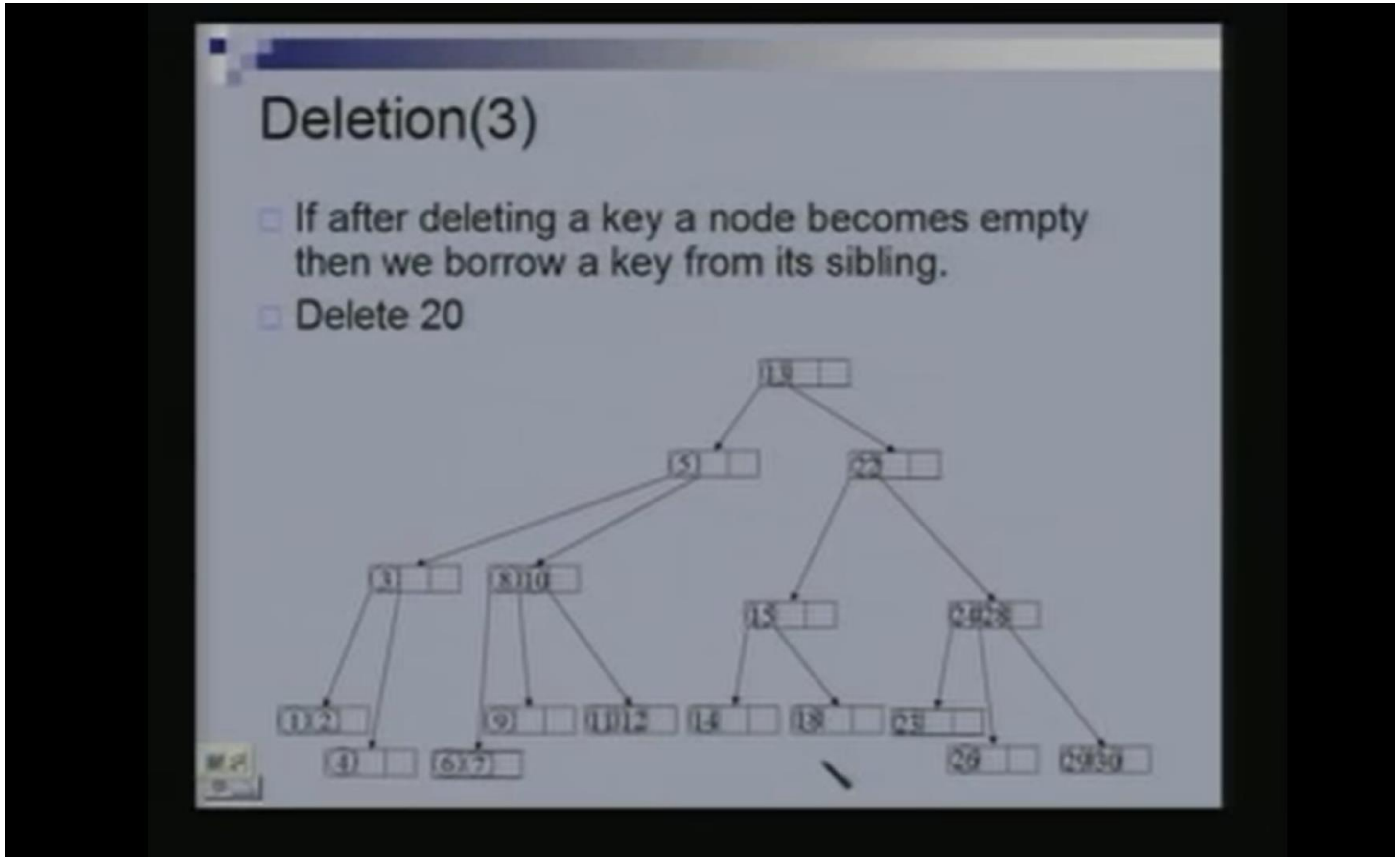
Deletion

- Case 3:- Nodes become empty





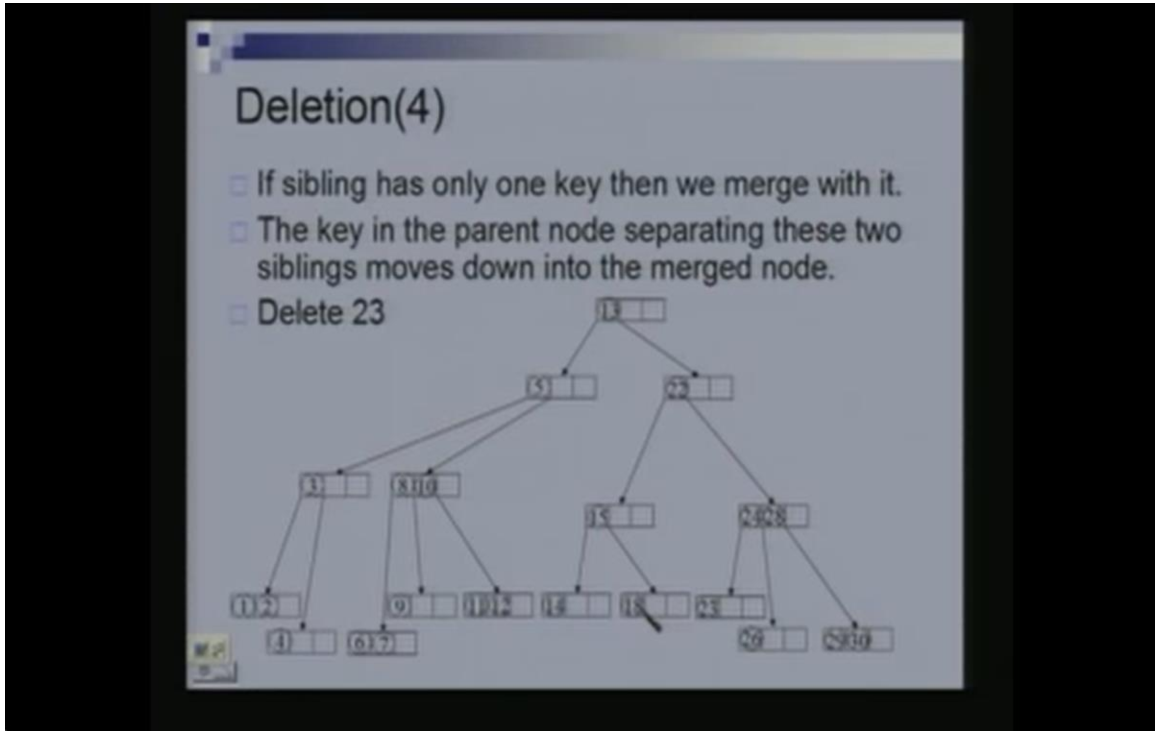
Deletion

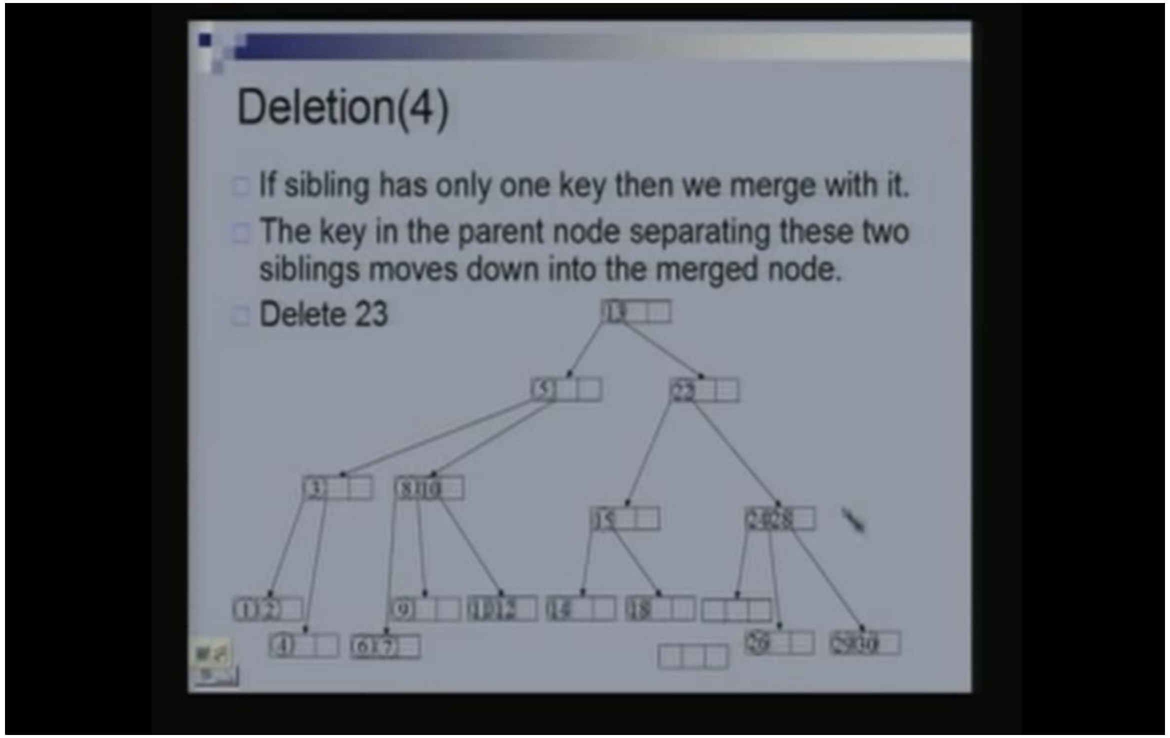


- Here we are borrowing a key from sibling and we cannot borrow 15 as it will not follow search property so we will do roatation as in avl tree so 15 goes up and 18 goes down.

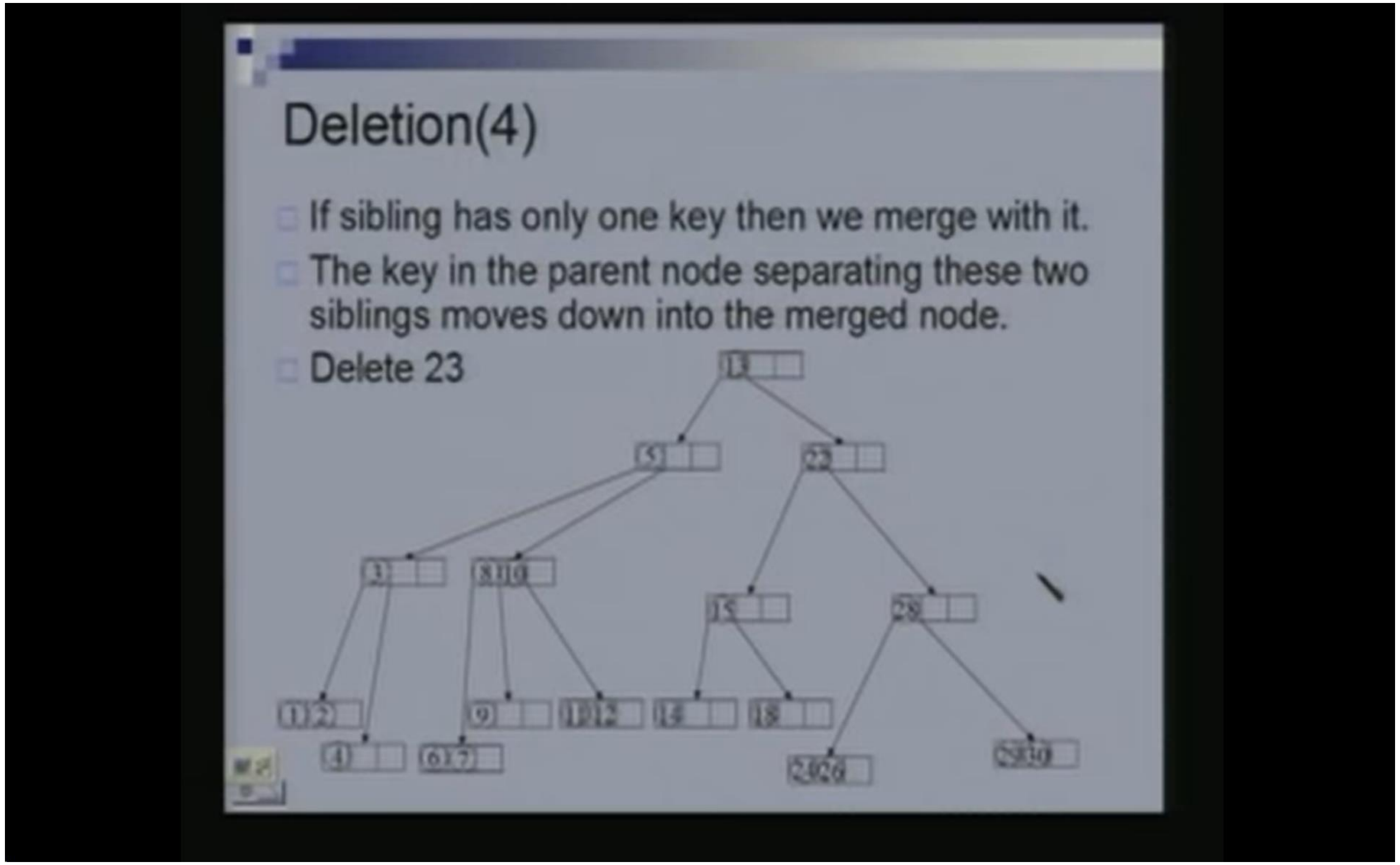
Deletion

- Case 4:- if we cannot borrow from sibling



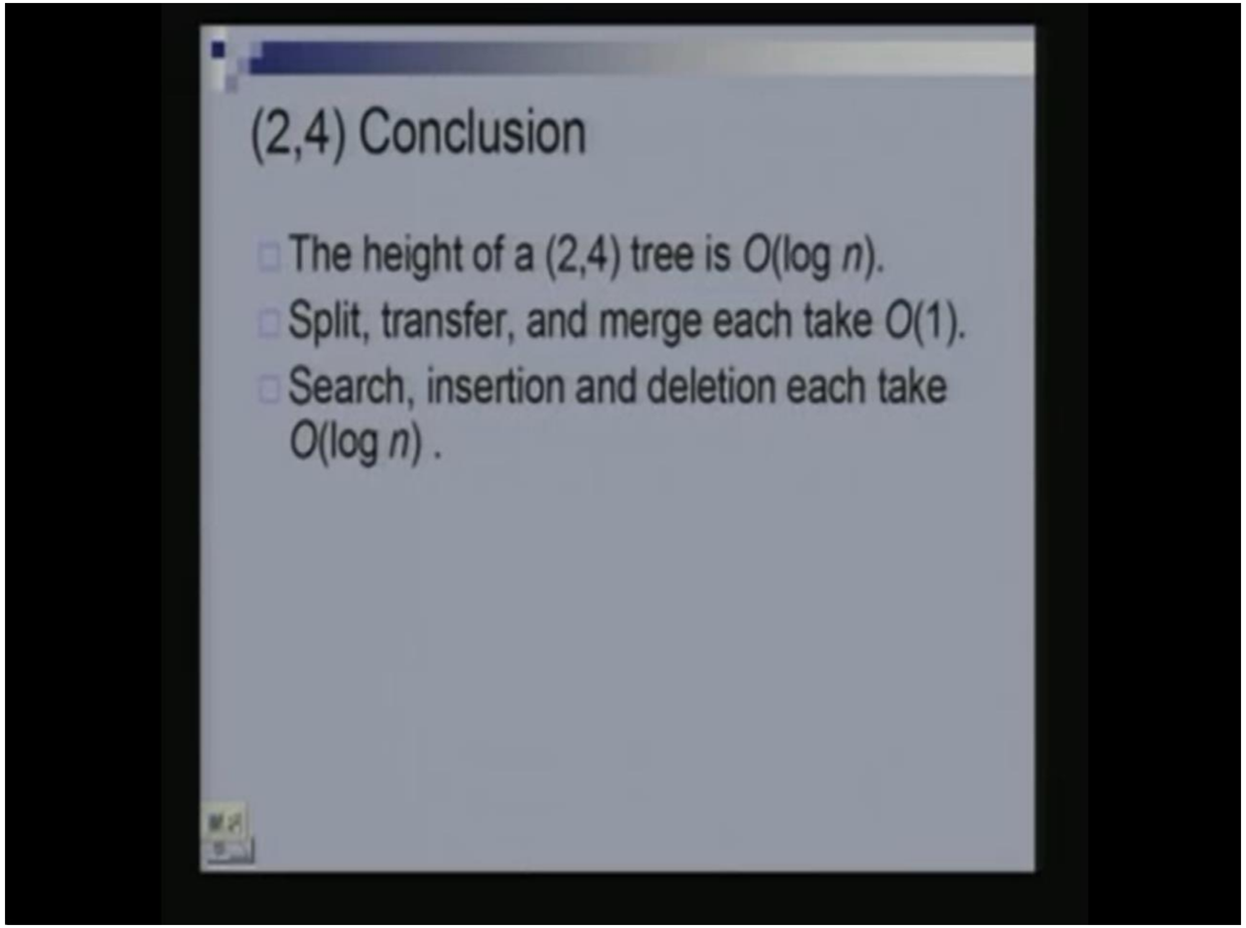


Deletion



- So here one key from parent comes down to the new node so there is 24 and 26 in the new nodeand the parent has one key with 2 children node.

Conclusion:-



- So, we studied this complex data structure because they are very connected to red black trees which are very useful.