

Example of B+ Deletion:

First, let's do an overview of B+ insertion and deletion:

Insertion

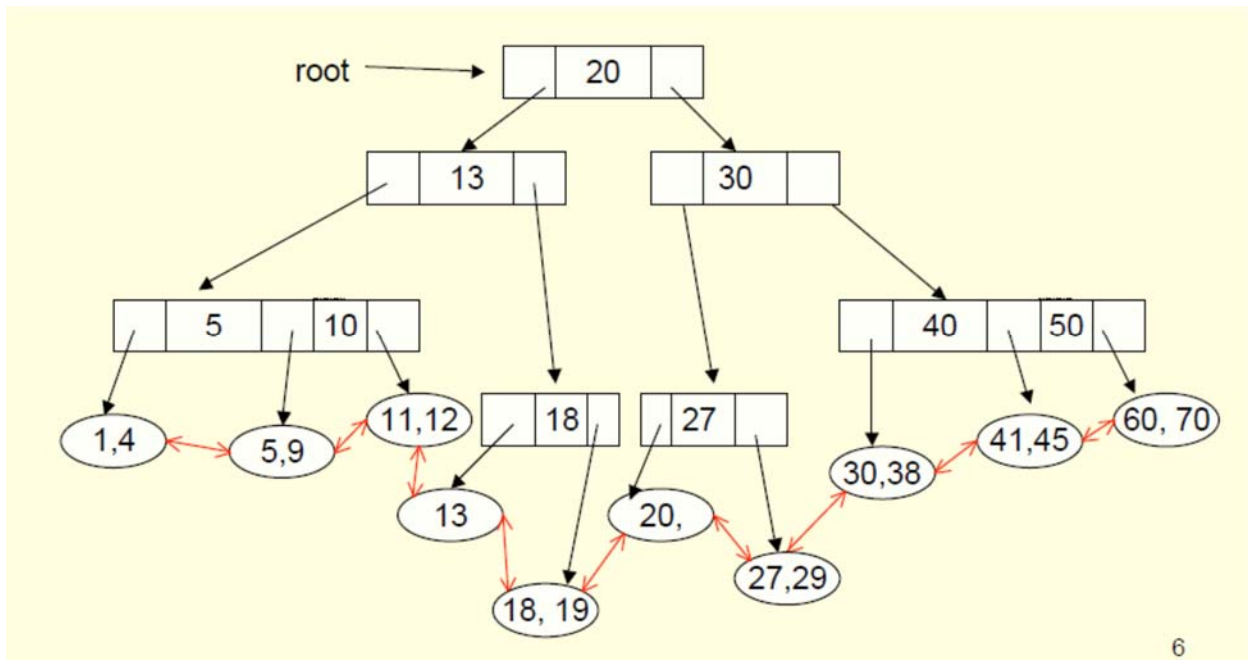
1. Find leaf which contains the value. If it is not full, then add value to leaf.
2. If leaf is full, spit in 2 nearly equal pieces, and promote a copy of the **least-valued key in the right split** to the parent. The difference is that the key value and record pointer stays at the leaf node.
3. Subsequent promotions at higher levels are identical to B trees. That is, one removes the center element, splits the node and promotes the center element up the tree. Only key values are being promoted.

Deletion

Find leaf with key value.

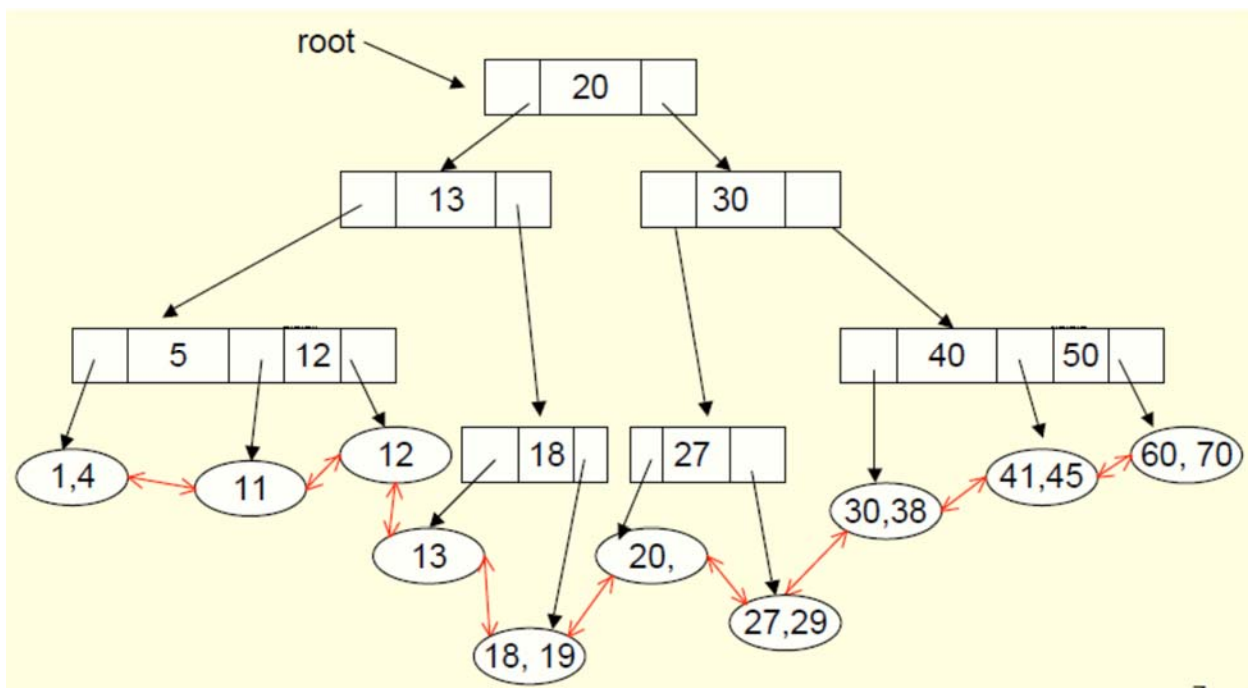
1. If leaf is more than 1/2 full, remove the key and record pointer.
2. If removing key reduces number of keys on the page below minimum number, one gets **underflow**. The tree needs to be restructured.
 - a. Look to the **adjacent** sibling leaves to see if they have a spare key. If so, **transfer enough records so both nodes wind up with same number of records** (this is to delay future underflows...). Update the key value of parent to reflect values in new leaves.
 - b. If neither sibling has records to spare, then merge node with one of the two siblings, and remove one of the parent's keys. This can cause **underflow at the parent!** When you merge two nodes, **remove the rightmost of the two nodes**, leaving only the left node, to keep the linked list structure at the bottom.
 - c. The above merging can result in underflow at the parent node. Recursive correct underflow at parent in the same way, merging nodes. **When you merge nodes, bring navigation key from grandparent down to parent.** If, at the end, the root has only one child, then remove the root node!

Example:

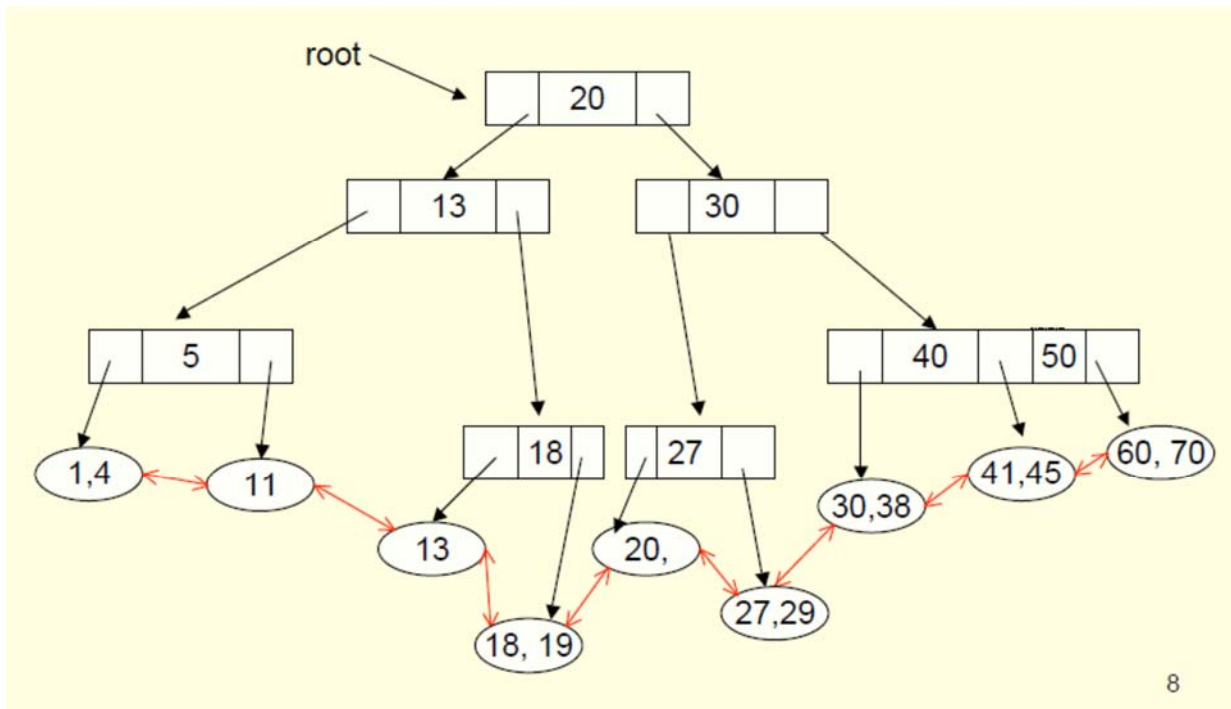


Delete 5 first: No problem, Leaf has extra key, delete it.

Delete 9: Leaf has underflow. Both neighbors have extra key. Borrow from right, update navigation pointer in parent.



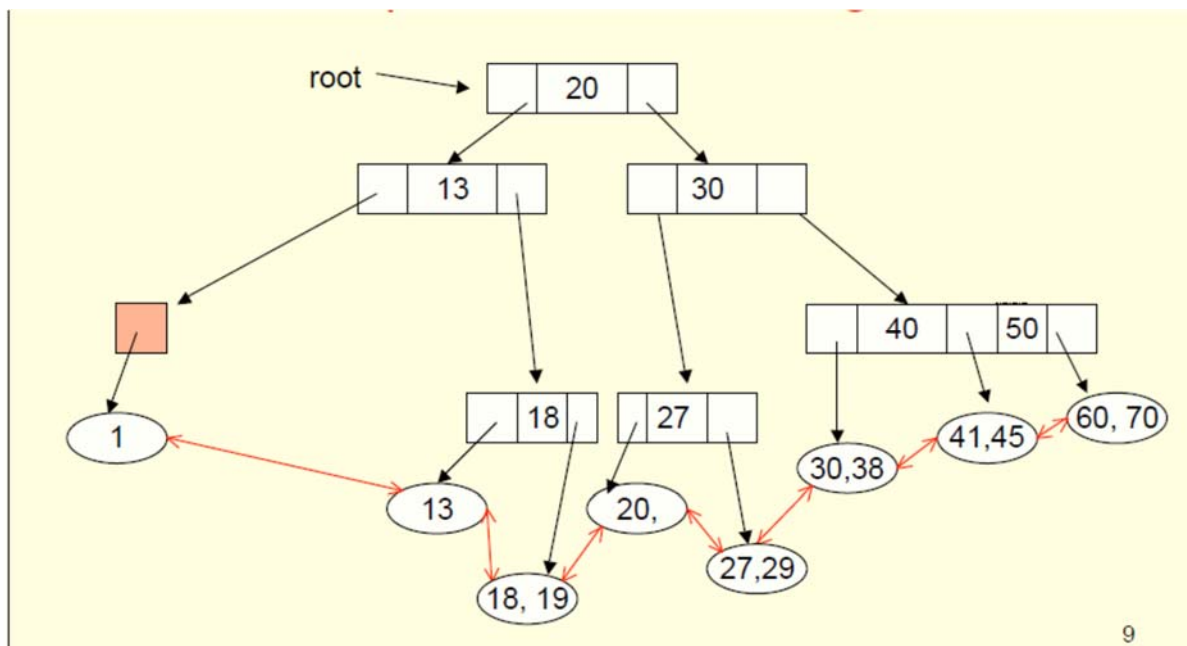
Delete 12: Another underflow. Neighbor has no extra keys, so merge. One child disappears, one key disappears from parent



8

Delete 4: Have extra key, so no problem. Just delete it.

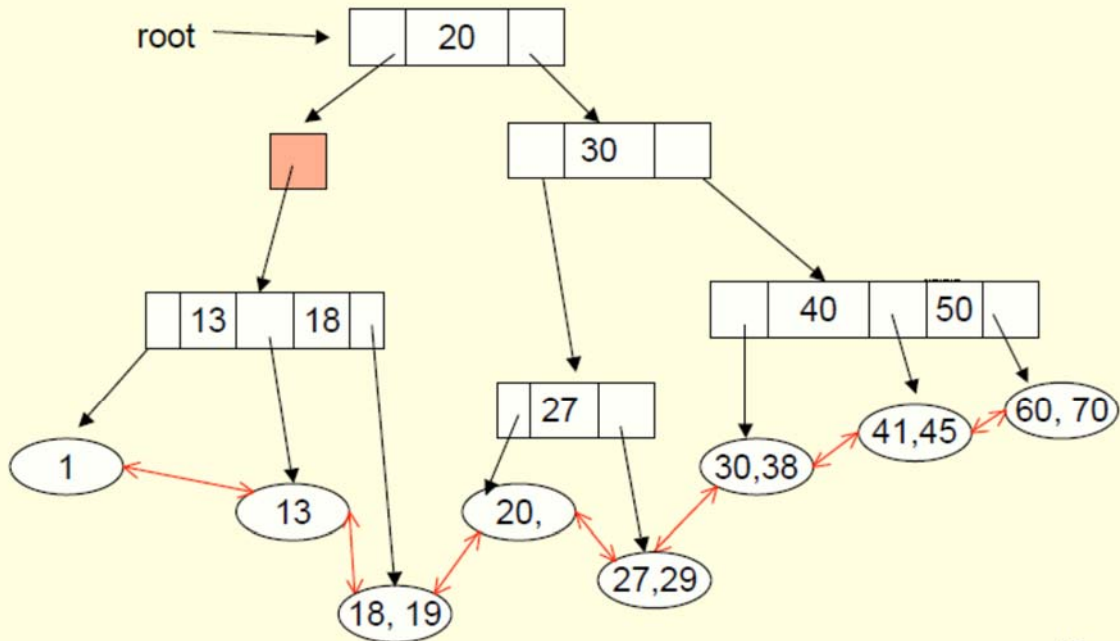
Delete 11. Underflow! No neighbor with extra keys...



9

Parent has underflow!!! Not enough children! Parent must merge. Bring down key from grandparent (13) into sibling for navigation.

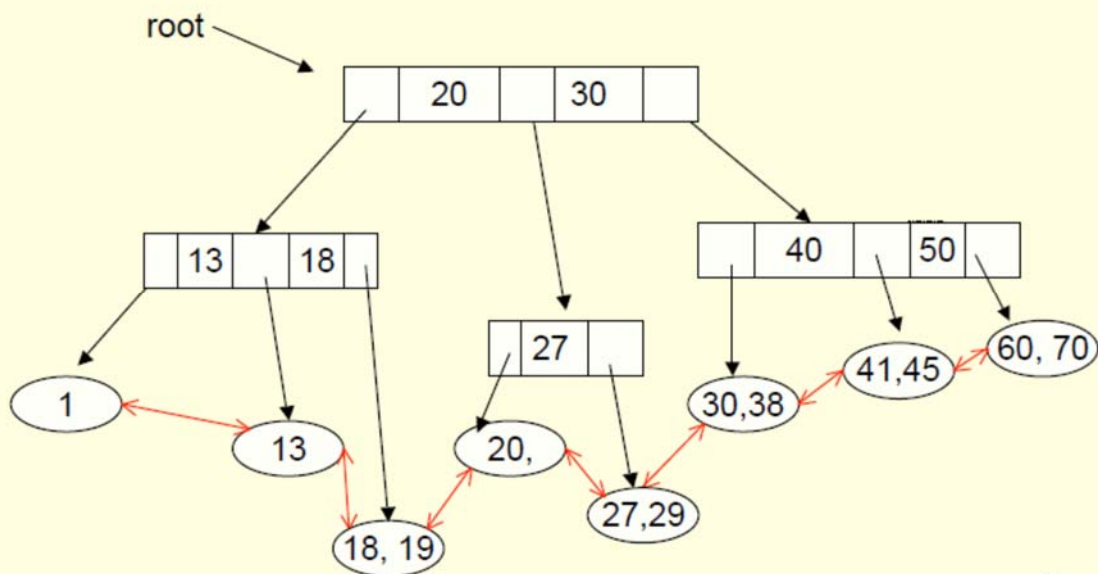
=>grandparent not full enough



10

Grandparents must merge. Bring down key from root for navigation.

Now root has underflow, so delete it!!!



11