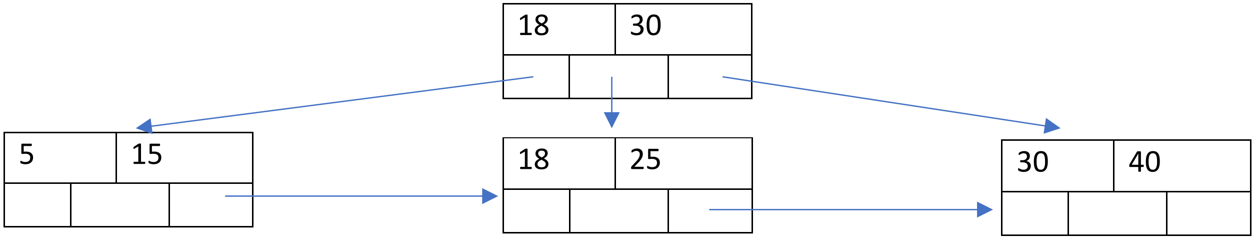
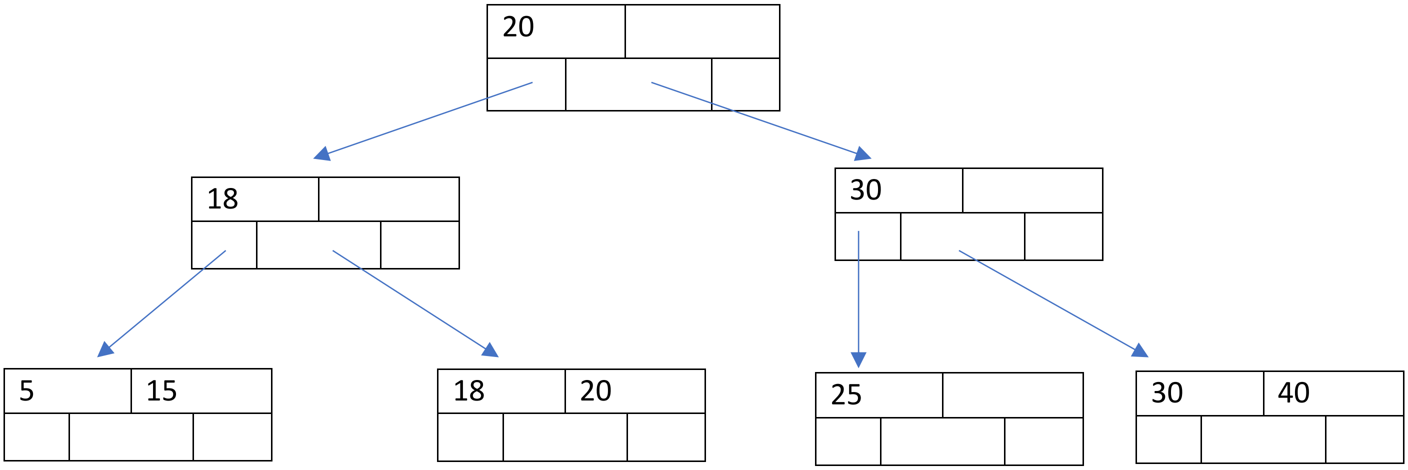
Group members:
Xiaoru Chen, chenxia2@oregonstate.edu, chenxia2
Yu-Chien Lin, linyuchi@oregonstate.edu, linyuchi

1.
a)



After Insert value 20:

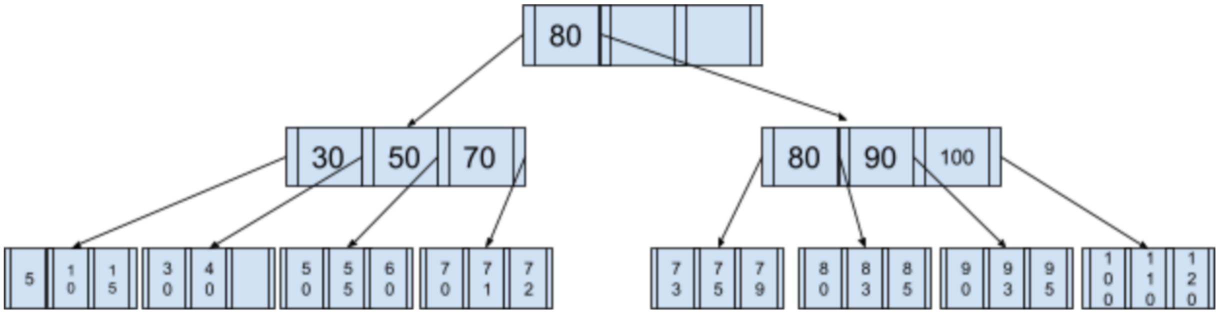


We need to insert 20 into the block [18, 25]. However, the space is not enough. We split the block into two blocks (18 and 25), and they should be 20's children. When we try to insert 20 into the block [18, 30], the space is not enough. Then we split the block into two blocks (18 and 30), and they should be 20's children at this time. Finally, make adjustment on the record pointers.

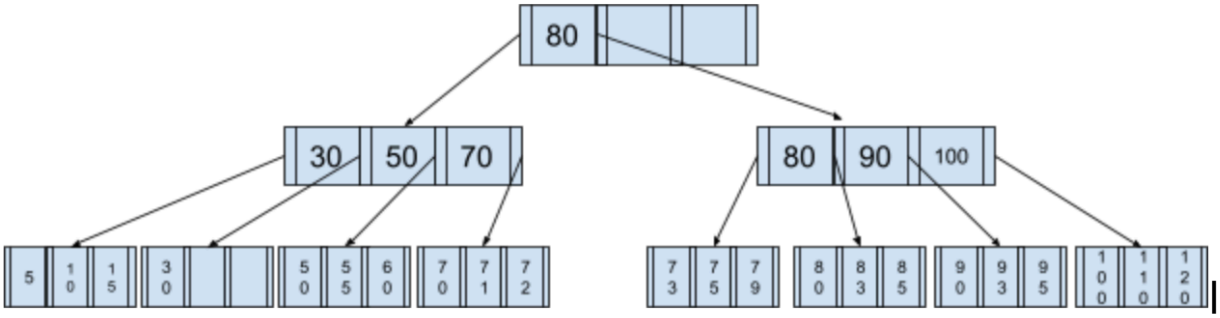
b)

We try to make the node in each block as much as 30 bytes.

In order to use “borrowing” to make distribution happen, the block that has value 40 is not full.

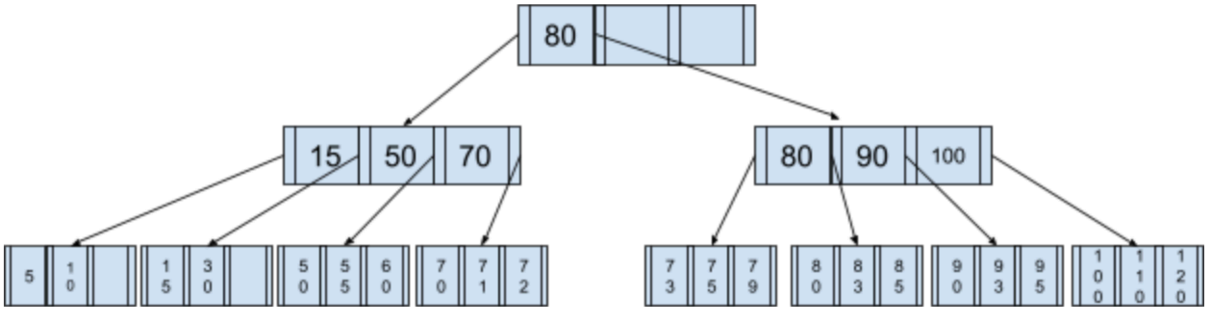


Delete value 40:



Then, k=30 cannot remain in the internal level, because each must be minimum 50% occupancy. Therefore, borrow from its sibling.

Then we got:



As shown above, index 30 also has to be rewrite to 15.

If all blocks of the original B+ Tree are full, then borrowing would not happen.

2.
Build a B+ tree, start with the internal nodes of age, from youngest to oldest. The leaf level contains all employee's ages. And each age should have a pointer to salary block. The salary block stores salary from lowest to highest.
Records are stored according to the index order, therefore, it should be clustered.

(Clustered B+ Tree index on <age, sal>, because in this query, records are sorted by age first and then by salary. Therefore, age equals to 20 should be clustered together).