B+ trees

Nodes type: Root, none-leaf (index, internal) and leaf (data).

Order: M (integer) controls the growth of B+ tree. Some time there are two orders M for none-leaf and L for leaf. In this document we will use M for both none-leaf and leaf.

Root node: in the beginning it is a leaf node after it is getting overflow becomes none-leaf node. When it is a leaf node it has a minimum 1 data and a maximum M data, when it is none-leaf it has a minimum 2 children and a maximum M children.

None-leaf node: Has a minimum M/2 (rounded up) children and maximum m children. The keys always should be Children - 1

Leaf node: Has a minimum M/2 (rounded up) data and maximum M data.

Inserting in B+ Tree:

Always add to the leaf node after going through the none-leaf nodes to find the appropriate leaf node.

After adding to the leaf node you should got one of the following cases

Case 1: leaf node is not overflow. (Normal case).

Case 2: Leaf node is overflow, do transfer for one key to sibling node and update the parent index node (if the parent index node not overflow), if cannot transfer do split for the leaf node.

How to split the leaf node:

You split the leaf node when it has more than M (the order) data.

1- First node gets M/2 data (rounded up).

2- Second node gets the remaining data.

3- Copy the smallest data in the second node into the parent none-leaf node.

Example 1:

Suppose we have M = 3 and we have the following leaf

8	10	12

Insert 15

8	10	12	15

Now the node has 4 data which is more than M (3), do the previous 3 steps:

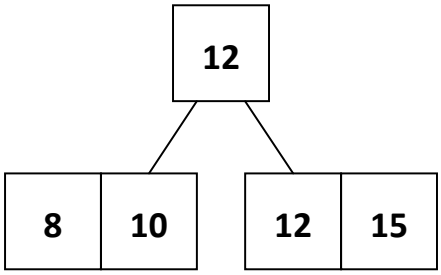
1- First node has M/2 = 3/2 = 1.5 = 2 (after rounding)

8	10

2- Second node has the remaining data.

12	15

3- Copy the smallest data into the parent (none-leaf node).



Example 2:

Suppose we have M = 4 and we have the following leaf

8	10	12	15

Insert 13

8	10	12	13	15

Now the node has 5 data which is more than M (4), do the previous 3 steps:

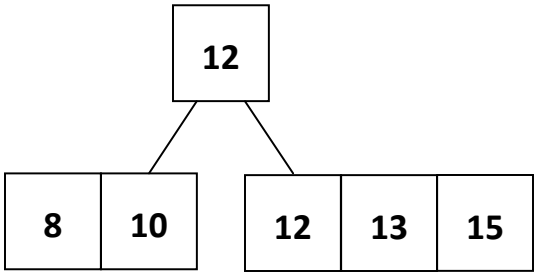
1- First node has M/2 = 4/2 = 2

8	10

2- Second node has the remaining data.

12	13	15

3- Copy the smallest data into the parent (none-leaf node).



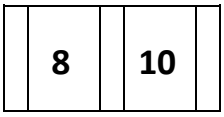
How to split the none-leaf node:

You split the none-leaf node when it has more than M children.

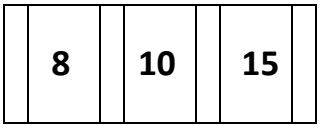
1- First node gets (M/2) keys with their children (rounded down).
2- Take the smallest key from the remaining and put it in the parent.
3- Second node gets the remaining keys with their children.

Example 1:

Suppose we have M = 3 and we have the following none-leaf node

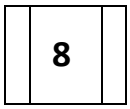


Insert 15

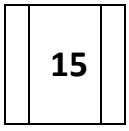


Now the node has 4 children which is more than M (3), do the previous 3 steps:

1- First node has (M/2) - 1 = 2 – 1 = 1 (after rounding down)

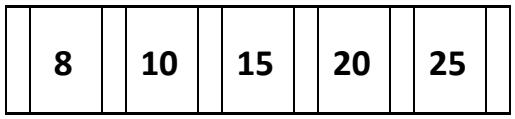


2- Copy the 10 key to the parent.
3- Second node has the remaining keys with their children.

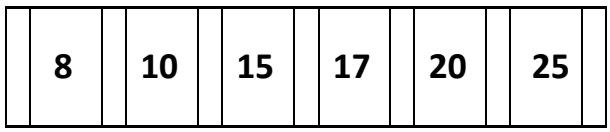


Example 2:

Suppose we have M = 6 and we have the following none-leaf node

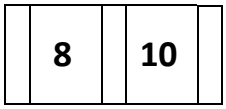


Insert 17



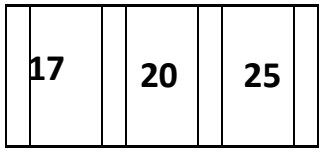
Now the node has 7 children which is more than M (6), do the previous 3 steps:

1- First node has (M/2) - 1 = 3 – 1 = 2



2- Copy the 15 key to the parent.

3- Second node has the remaining keys with their children.



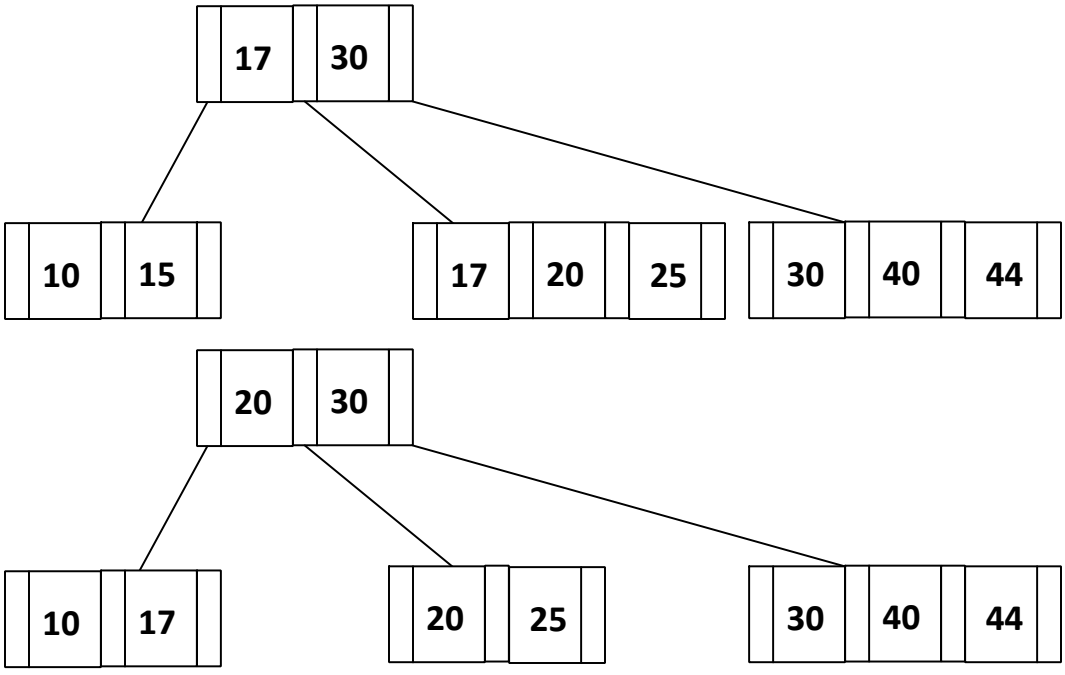
Delete

1- **Search for the data through the none-leaf nodes, find it, and delete the data from leaf node.**
2- **Check the leaf node; it should have at least M/2 data (round up).**
3- **If the leaf node has less then M/2 borrow from its sibling, if the sibling has more than M/2 data.**
4- **If the sibling has M/2 data exactly, than you cannot borrow, merge the two leaf nodes, and adjust the parent none-leaf node.**

Example 1(Borrow, redistribute) Suppose

M = 3

Delete 15



Example 2(Merge)

Suppose M = 3

Delete 15