

B+ Tree

B+ Tree is an extension of B Tree which allows efficient insertion, deletion and search operations.

In B Tree, Keys and records both can be stored in the internal as well as leaf nodes. Whereas, in B+ tree, records (data) can only be stored on the leaf nodes while internal nodes can only store the key values.

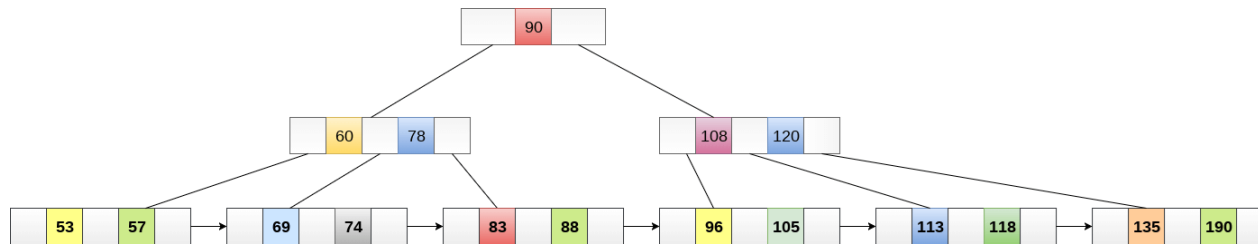
The leaf nodes of a B+ tree are linked together in the form of a singly linked lists to make the search queries more efficient.

An advertisement for the WE app. It features a man in a blue shirt performing a handstand on a blue mat. The text in Arabic promotes the app, mentioning a 300x increase in likes and a 100x increase in followers. The WE logo is visible in the top right corner of the ad. Below the ad, there is a small icon of a person and the text 'WE'.

B+ Tree are used to store the large amount of data which can not be stored in the main memory. Due to the fact that, size of main memory is always limited, the internal nodes (keys to access records) of the B+ tree are stored in the main memory whereas, leaf nodes are stored in the secondary memory.



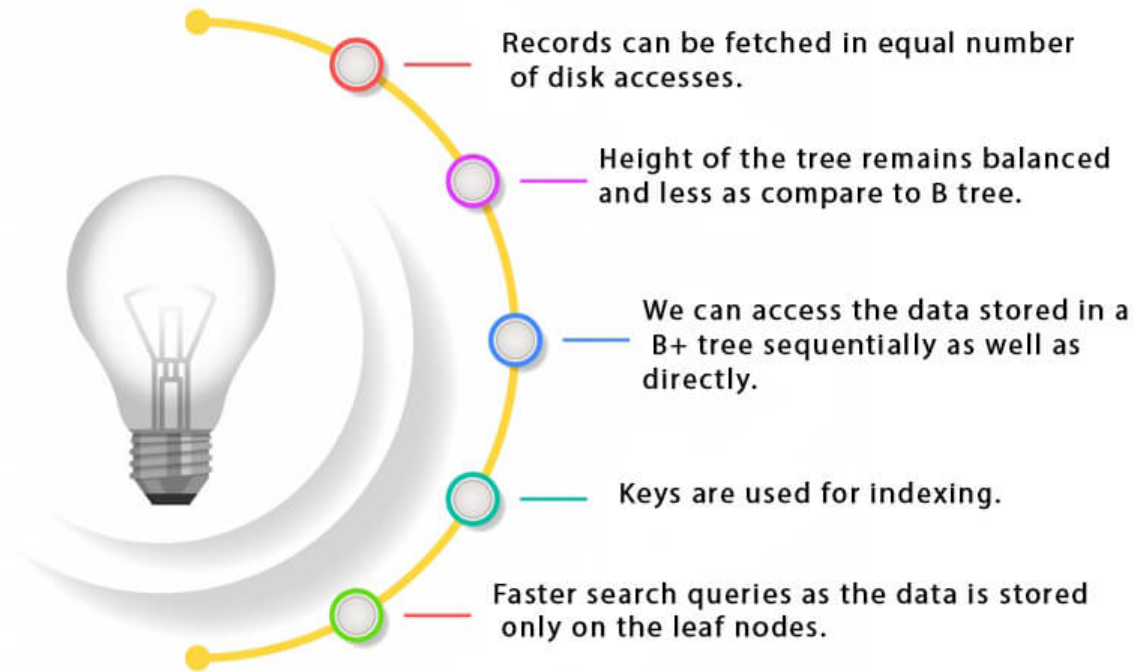
The internal nodes of B+ tree are often called index nodes. A B+ tree of order 3 is shown in the following figure.



Advantages of B+ Tree

1. Records can be fetched in equal number of disk accesses.
2. Height of the tree remains balanced and less as compare to B tree.
3. We can access the data stored in a B+ tree sequentially as well as directly.
4. Keys are used for indexing.
5. Faster search queries as the data is stored only on the leaf nodes.

Advantages of B+ Tree



B Tree VS B+ Tree

SN	B Tree	B+ Tree
1	Search keys can not be repeatedly stored.	Redundant search keys can be present.
2	Data can be stored in leaf nodes as well as internal nodes	Data can only be stored on the leaf nodes.



3	Searching for some data is a slower process since data can be found on internal nodes as well as on the leaf nodes.	Searching is comparatively faster as data can only be found on the leaf nodes.
4	Deletion of internal nodes are so complicated and time consuming.	Deletion will never be a complexed process since element will always be deleted from the leaf nodes.
5	Leaf nodes can not be linked together.	Leaf nodes are linked together to make the search operations more efficient.

Insertion in B+ Tree

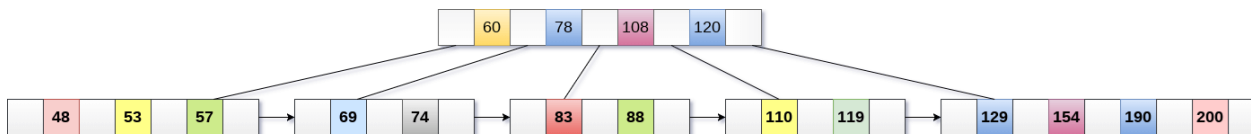
Step 1: Insert the new node as a leaf node

Step 2: If the leaf doesn't have required space, split the node and copy the middle node to the next index node.

Step 3: If the index node doesn't have required space, split the node and copy the middle element to the next index page.

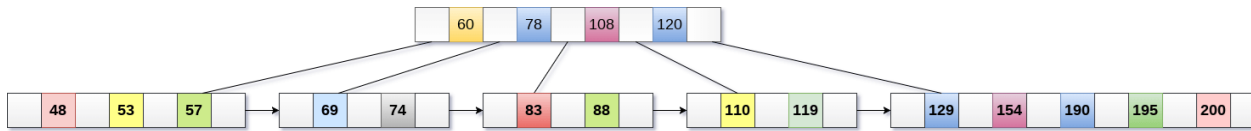
Example :

Insert the value 195 into the B+ tree of order 5 shown in the following figure.

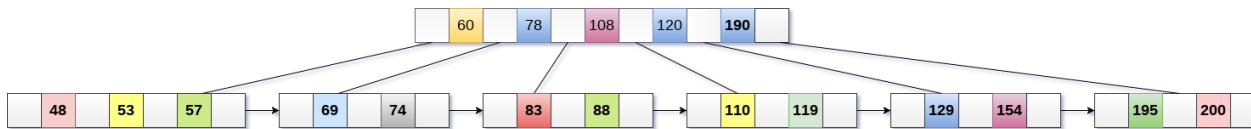


195 will be inserted in the right sub-tree of 120 after 190. Insert it at the desired position.

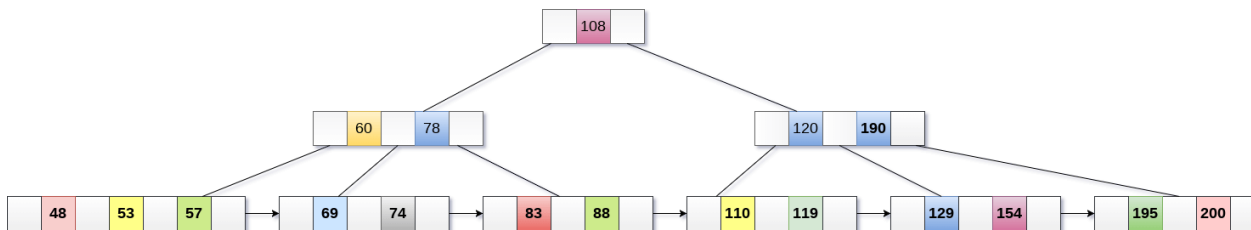




The node contains greater than the maximum number of elements i.e. 4, therefore split it and place the median node up to the parent.



Now, the index node contains 6 children and 5 keys which violates the B+ tree properties, therefore we need to split it, shown as follows.



Deletion in B+ Tree

Step 1: Delete the key and data from the leaves.

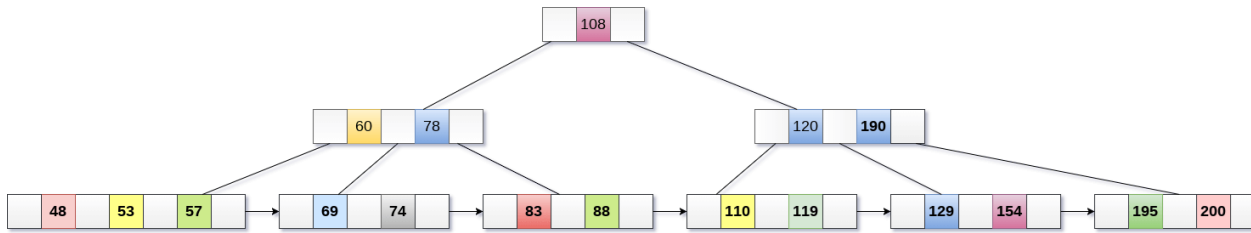
Step 2: if the leaf node contains less than minimum number of elements, merge down the node with its sibling and delete the key in between them.

Step 3: if the index node contains less than minimum number of elements, merge the node with the sibling and move down the key in between them.

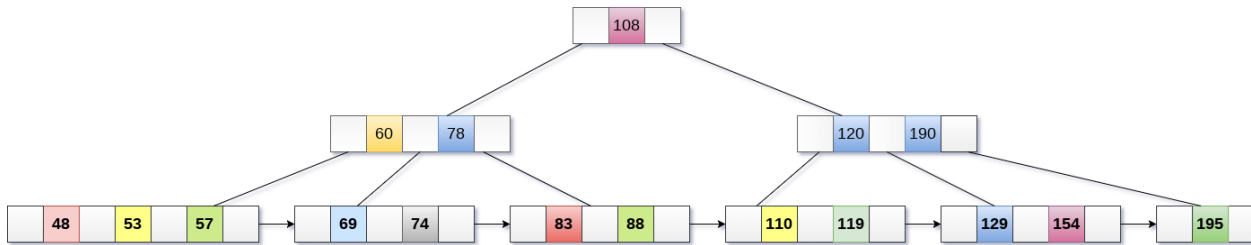
Example



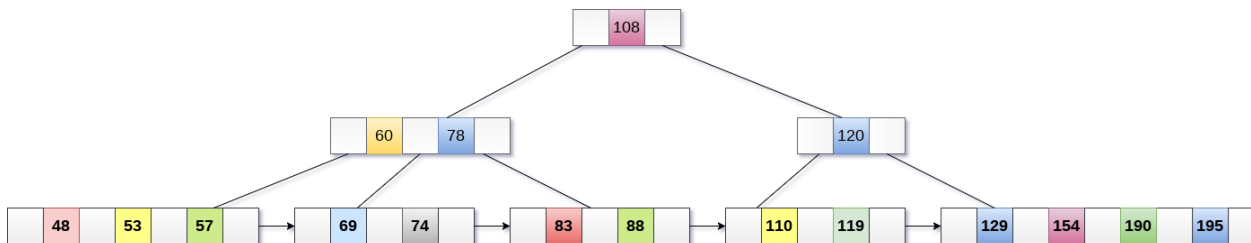
Delete the key 200 from the B+ Tree shown in the following figure.



200 is present in the right sub-tree of 190, after 195. delete it.



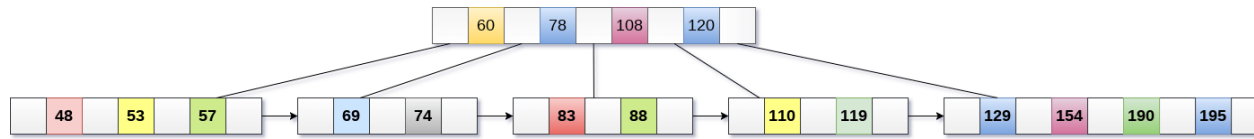
Merge the two nodes by using 195, 190, 154 and 129.



Now, element 120 is the single element present in the node which is violating the B+ Tree properties. Therefore, we need to merge it by using 60, 78, 108 and 120.

Now, the height of B+ tree will be decreased by 1.





← prev

next →

Help Others, Please Share



Learn Latest Tutorials



GDB

GDB



Fuzzy Logic

Fuzzy Logic



DHTML

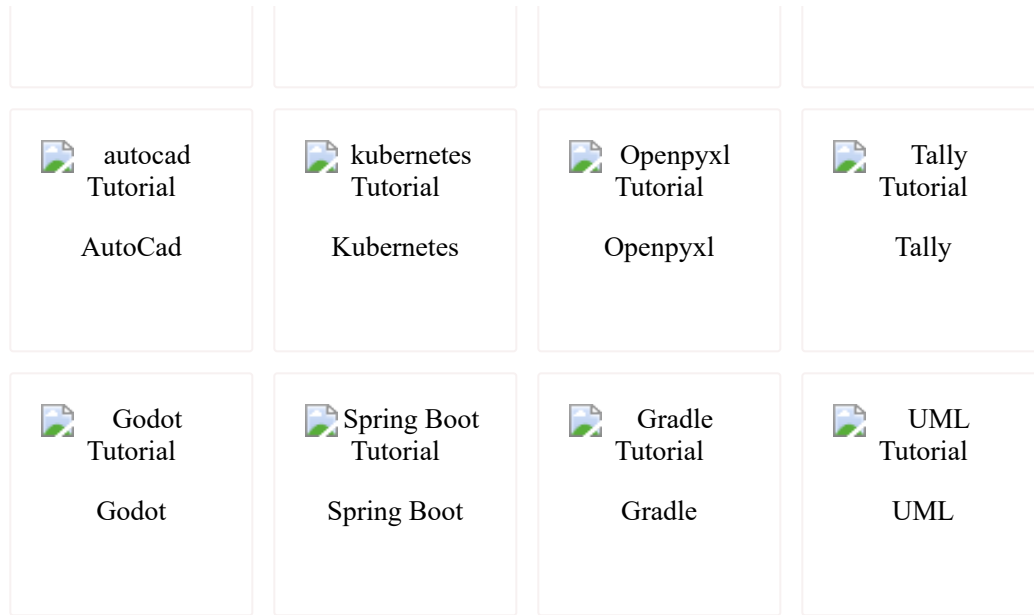
DHTML



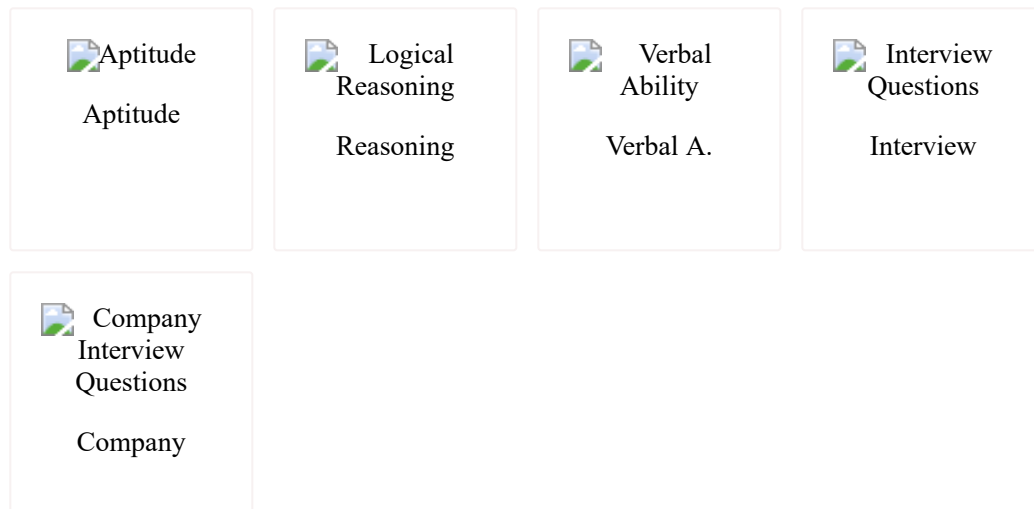
Google
Classroom

Classroom



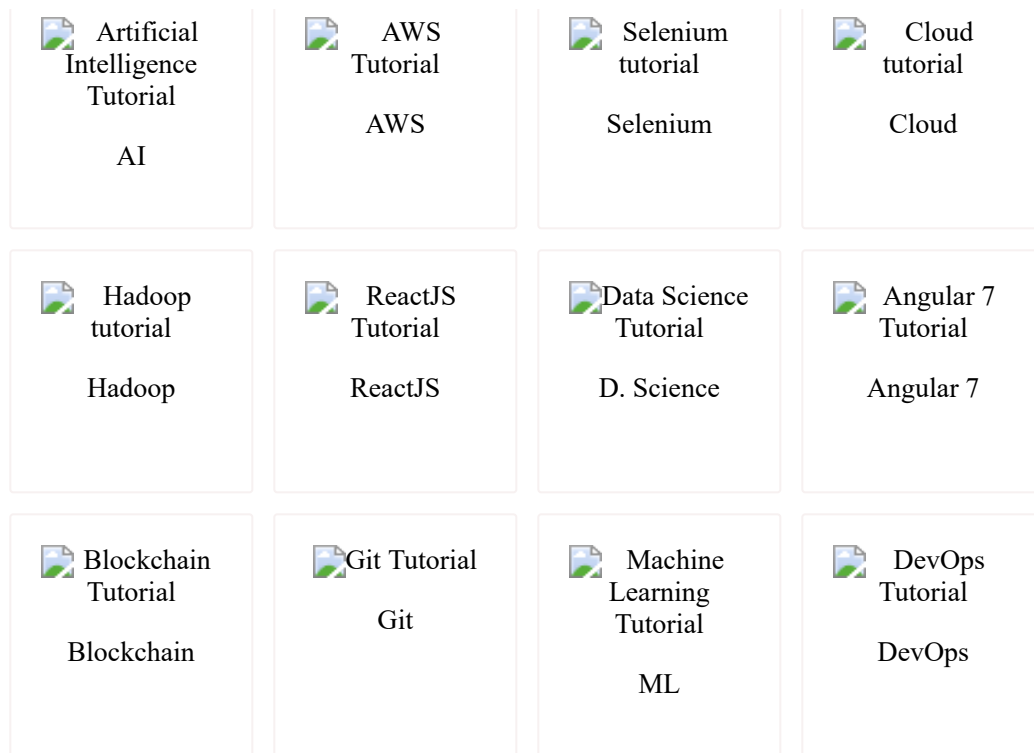


Preparation

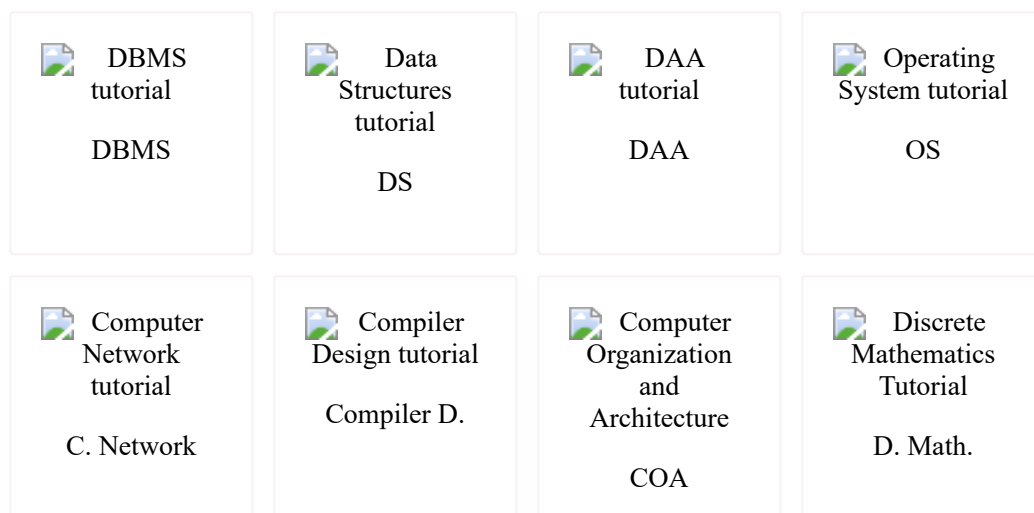


Trending Technologies





B.Tech / MCA





Ethical
Hacking
Tutorial

E. Hacking



Computer
Graphics
Tutorial

C. Graphics



Software
Engineering
Tutorial

Software E.



html tutorial
Web Tech.



Cyber
Security
tutorial

Cyber Sec.



Automata
Tutorial

Automata



C Language
tutorial

C



C++ tutorial
C++



Java tutorial
Java



.Net
Framework
tutorial

.Net



Python
tutorial
Python



List of
Programs
Programs



Control
Systems
tutorial

Control S.



Data Mining
Tutorial

Data Mining

