**Project Report**

**Dictionary**

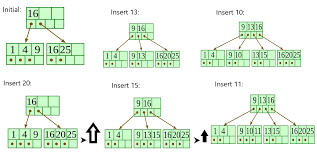
Introduction:

We have put together a dictionary of 400 words approx. as our semester project of Data Structures. In which we use B+ Tree as the core topic of the project.

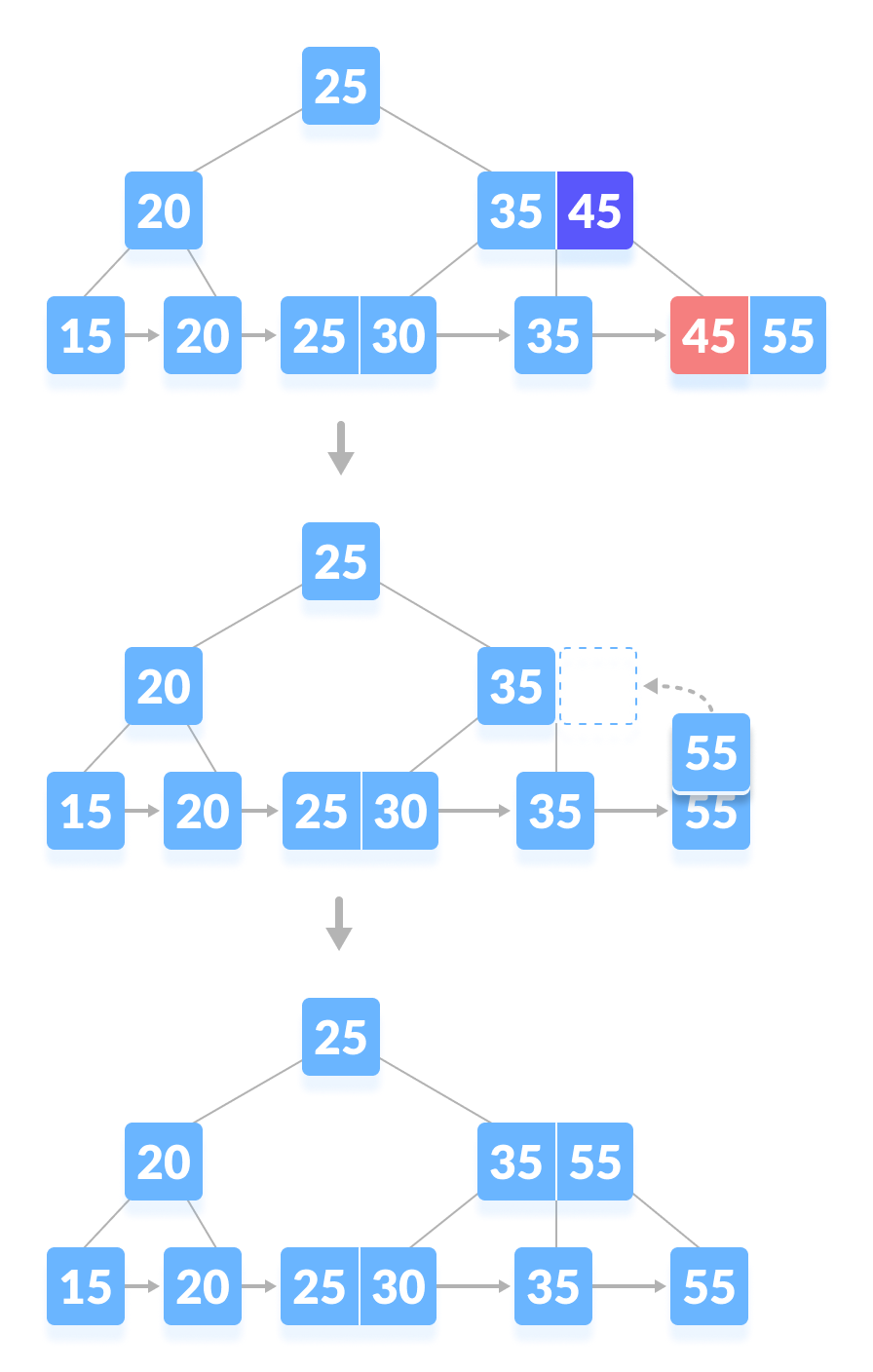
B+ Tree:

B+ Tree is an m-ary tree with one or more number of children per node. In B+ tree the link or address to the data is stored in the internal node while the data is stored in leaf nodes.

Insertion in B+ Tree:



Deletion in B+ Tree:



Why B+ Tree ?

B+ Tree stores the data pointers only at the leaf nodes of the tree, which makes search process more accurate and faster in B+ Tree.

|  |  |  |
| --- | --- | --- |
| **Alogrithm** | **Average** | **Worst Case** |
| **Space** | O(n) | O(n) |
| **Search** | O(log n) | O(log n + log L) |
| **Insert** | O(log n) | O(M\*log n + log L) |
| **Delete** | O(log n) | O(M\*log n + log L) |

Applications of B+ Tree:

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

Project Specification:

Dictionary conatins word’s meaning, synonyms and antonyms. A word can be add up to dictionary, a word can be searched and deleted from dictionary and whole dictionary can also be viewed.

Topics Covered:

* B+ Tree
* Filing
* Recursion
* Dynamic Memory Allocation
* Some OOP concepts.