Project 2: Implementing B+ Trees

Inserting a node in B+ Tree involves handling of several cases like if the node to be inserted in the tree will belong to a leaf node OR if it belongs to an internal node. After determining if it belongs to leaf or internal level, we need to determine if it has an available location where it can be stored otherwise if that node is full then we split that node using either spiltL (i.e. split leaf) or splitI (i.e. split internal) node. We might need to cascade successive splits to one level up till it forms new root node. After splitting the correct node, we call wedge method to insert it into the node. Again, it could be a wedge on either a leaf node or an internal node. There were several rules to determine new parent node after splitting which depend on if the node to be split on is at a leaf or internal level. There are several other utility methods as well, such as, firstKey(), lastKey() and headMap(), tailMap(), subMap() etc which support point queries. Alos, we implemented an entrySet() method which works as iterator for BPTreeMap class.