Data structures used:

1. B+ tree:
   tree having height of log n(base d) where d is the order of the b+ tree and n is the number of values inserted.
   ```
   typedef struct node
   {
           int keys[M-1];
           int n;
           struct node *p[M];
   }node;
   ```
   where M is the order of b+ tree,
   keys[M-1] store the data,
   n works as a counter of data inserted,
   p[M] is the pointer array for linking to intermediate nodes.

2. AVL tree:
   tree having height of log n(base 2) where n is the number of nodes in the tree.
   ```
   typedef struct node
   {
           int key;
           struct node *left;
           struct node *right;
           int height;
   }node;
   ```
   where key holds the valueof the data to be stored,
   left is the pointer of the left child,
   right is the pointer of the right child and
   height is the height of the node.