



DEPARTMENT OF INFORMATION TECHNOLOGY

Academic Year: 2023 – 24

COURSE CODE: DJS22ITL303

CLASS: S. Y. B. Tech. Sem III (I1)

COURSE NAME: Database Management Systems

SAP ID:60003220045

NAME: Anish Sharma

DATE: 3rd December 2023

EXPERIMENT NO:10

To Implement Binary Trees

CODE:

```
class BTreeNode {
    int[] keys;
    int t; // Minimum degree
    BTreeNode[] children;
    int n; // Current number of keys
    boolean leaf;

    BTreeNode(int t, boolean leaf) {
        this.t = t;
        this.leaf = leaf;
        keys = new int[2 * t - 1];
        children = new BTreeNode[2 * t];
        n = 0;
    }

    void insertNonFull(int key) {
        int i = n - 1;
        if (leaf) {
            while (i >= 0 && key < keys[i]) {
```



```
    keys[i + 1] = keys[i];  
    i--;  
}  
keys[i + 1] = key;  
n++;  
} else {  
    while (i >= 0 && key < keys[i]) {  
        i--;  
    }  
    if (children[i + 1].n == 2 * t - 1) {  
        splitChild(i + 1, children[i + 1]);  
        if (key > keys[i + 1]) {  
            i++;  
        }  
    }  
    children[i + 1].insertNonFull(key);  
}  
}
```

```
void splitChild(int i, BTreeNode y) {  
    BTreeNode z = new BTreeNode(y.t, y.leaf);  
    z.n = t - 1;  
  
    for (int j = 0; j < t - 1; j++) {  
        z.keys[j] = y.keys[j + t];  
    }  
  
    if (!y.leaf) {  
        for (int j = 0; j < t; j++) {
```



```
        z.children[j] = y.children[j + t];
    }
}

y.n = t - 1;

for (int j = n; j >= i + 1; j--) {
    children[j + 1] = children[j];
}
children[i + 1] = z;

for (int j = n - 1; j >= i; j--) {
    keys[j + 1] = keys[j];
}
keys[i] = y.keys[t - 1];

n++;
}
```

```
void traverse() {
    int i;
    for (i = 0; i < n; i++) {
        if (!leaf) {
            children[i].traverse();
        }
        System.out.print(keys[i] + " ");
    }

    if (!leaf) {
```



```
        children[i].traverse();  
    }  
}  
}
```

```
class BTree {  
    private BTreeNode root;  
    private int t;  
  
    BTree(int t) {  
        this.t = t;  
        root = new BTreeNode(t, true);  
    }
```

```
    void insert(int key) {  
        if (root.n == 2 * t - 1) {  
            BTreeNode s = new BTreeNode(t, false);  
            s.children[0] = root;  
            s.splitChild(0, root);  
  
            int i = 0;  
            if (s.keys[0] < key) {  
                i++;  
            }  
            s.children[i].insertNonFull(key);  
            root = s;  
        } else {  
            root.insertNonFull(key);  
        }  
    }
```



```
}

void traverse() {
    if (root != null) {
        root.traverse();
    }
}

}

class Main {
    public static void main(String[] args) {
        int[] sequence = { 120, 2, 45, 201, 42, 78, 350, 401, 50, 135, 88, 71 };
        int degree = 3; // Degree of the B-tree

        BTree bTree = new BTree(degree);

        for (int key : sequence) {
            bTree.insert(key);
        }

        System.out.println("B-tree traversal:");
        bTree.traverse();
    }
}
```

OUTPUT:

```
B-tree traversal:
2 42 45 50 71 78 88 120 135 201 350 401
```