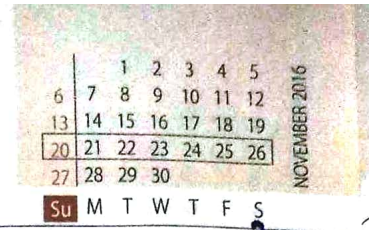
22
WK 48 - 327-039

NOVEMBER 2016
TUESDAY



ADS

IBM19CS401
Arbaz Ahmad

11/11/20/wed

B-Tree Insertion

```
class BTreeNode {
    int *keys;
    int t, n;
    BTreeNode **C;
    bool leaf
};
BTreeNode(int t1, bool leaf)
{
    int t = t1
    keys = new int [2*t-1];
    C = new BTreeNode *(2*t];
    n = 0;
}
void insert(int k)
{
    if (root == NULL) {
        root = new BTreeNode(t, true);
        root->keys[0] = k;
    } else {
        root->n = 1;
        if (root->n == 2*t-1) {
```

```
BTreeNode *S = new BTree(t, false);
S->C[0] = root
S-> splitchild(0, root);
int i=0;
if (s->keys[0]<k)
    i++
S->C[i] -> insertNonFull(k);
root = s;
} else {
    root -> insertNonFull(k);
}
}

insertNonFull(int k) {
int i=n-1;
if (leaf == true) {
while(i>=0 && keys[i]>k) {
keys[i+1] = keys[i];
i--;
}
keys[i+1] = k;
n = n+1;
} else { while(i>=0 && keys[i]>k)
i--;
```

```
if (C[i+1]->n == 2*t-1)
{
    splitChild (i+1, C[i+1]);
    if (keys[i+1] < k)
        i++;
}
C[i+1] -> insertNonfull (k);
}
}

* 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->key[j+t];
    if (y->leaf == false){
        for (j=0; j<t; j++)
            z->C[j] = y->C[j+t];
    }
    y->n = t-1;
    for (j=n; j>=i+1; j--)
        C[j+1] = C[j];
```

```
c[i+1] = z
for (j = n-1 ; j >= i ; j--)
    keys[j+1] = keys[j];
keys[i] = y->keys[i];
n = n+1;
}
```