

11/11/2020  
Wednesday.

ADS

Lab-6 Writing

B-trees

Arjun A.S  
(BM1815019)

class node

```
{
    int *data;
    int
    node **child;
    int n;
    bool leaf;
public:
    node(int, bool);
    void insertionnonfull(int item);
    void splitchild(int i, node * y);
    void traverse();
}
```

friend class Btree;

class Btree.

```
{
    node * root;
    int m;
public:
    Btree(int m)
    {
        root = NULL;
        m = m;
    }
    void traverse()
    {
        if (root != NULL)
            root->traverse();
    }
    void insertion(int item);
}
```

node :: node(int m, bool leaf)

```
{
    m = m;
    leaf = leaf;
    data = new int[2*m-1];
    child = new node*[2*m];
    n = 0;
}
```

Arjun A.S

Abhinav S.  
19m18csc019

void Btree :: insertion (int item)

{

if (root == NULL)

{  
root = new node(m, true);  
root → data[0] = item;  
root → n = 1;

}

else

{

if (root → n == 2<sup>m</sup> - 1)

{  
node \* s = new node(m, false);  
s → child[0] = root;  
s → splitchild(0, root);

int i = 0;

if (s → data[i] < item)

i++;

s → child[i] → insertionnonfull(item);  
root = s;

}

else

root → insertionnonfull(item);

}

}

void node :: insertionnonfull (int item)

{

int i = n - 1;

if (leaf == true)

{ while (i ≥ 0 && data[i] > item)

{ data[i+1] = data[i];

i--;

}

data[i+1] = item;

n = n + 1;

}

else

{

while (i ≥ 0 && data[i] > item)

i--;

if (child[i+1] → n == 2<sup>m</sup> - 1)

{ splitchild(i+1, child[i+1]);

if (data[i+1] < item)  
i++;

Arjun A.S  
18m18c5019

child[i+1] → insertionnonfull(item);

}  
void  
{

node:: splitchild (int i, node \*y)

node \*z = new node (y->m, y->leaf);

z->n = m-1;

for (int j=0; j < m; j++)  
z->~~data~~<sup>data</sup>[j] = y->~~child~~<sup>data</sup>[j+m];

if (y->leaf == false)

{  
for (int j=0; j < m; j++)  
z->child[j] = y->child[j+m];

}

y->n = m-1;

for (int j=n; j >= i+1; j--)  
child[j+1] = child[j];

child[i+1] = z;

for (int j=n-1; j >= i; j--)  
data[j+1] = data[j];

data[i] = y->data[m-1];

n=n+1;

}

Arjun A.S