SOUMYA.D.U.
1BM19CS407

Freedom Gold

Date:
Page:

Inserting a Key on a B-tree

```
//Insert the node
void BTree :: insert (int K)
{
  if (root == NULL)
  {
    root = newNode (t, true);
    root -> keys [0] = K;
    root -> n = 1;
  }
  else
  {
    if (root -> n == 2 * t - 1)
    {
      Node * s = new Node (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);
  }
}
//Insert non full condition
void Node :: 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);
}
}
//Split child
void Node::splitChild(int i, Node *y)
{ Node *z = new Node(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 == false)
{ for (int j = 0; j < t; j++)
    z->C[j] = y->C[j+t];
}
```

```
y->n = t-1;
for (int j=n; j>=i+1; j--)
  c[j+1] = c[j];

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

keys[i] = y->keys[t-1];
  n=n+1;
}
```