Prathik Karanth
1BM18CS139

B-Tree insertion

```
void insert(int k)
{ if (root == NULL)
  { root = newNode(t, true);
    root->key[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->key[0] < k)
        i++;
      s->C[i]->insertNonFull(k);
      root = s;
      } else
        root->insertNonFull(k);
    }
  }
void insertNonFull(int k)
{ int i=n-1;
  if (leaf == true)
  { while (i>=0 && key[i]>k)
    { key[i+1] = key[i];
      i--;
    }
    key[i+1] = k;
    n = n+1;
  }
```

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