2-3 Trees

Krishnaprasad V
1BM19CS047
5A

Insert

```
void insert (int k)
{   if (!root)
    {  root = new TreeNode (true);
       root -> keys [0] = k;
       root -> n = 1;
    }
    else
    {  if (root -> n == 3)
       {  TreeNode *s = new TreeNode (false);
          s -> child [0] = root;
          s -> splitChild (0, root);
          int i = 0;
          if (s -> keys [0] < k)  ++i;
          s -> child [i] -> insertNonFull (k);
          root = s;
       }
       else
          root -> insertNonFull (k);
    }
}
```

Delete

```
void delete (int k)
{   int idx = find (k)
    if (idx < n && keys[idx] == k)
    {  if (leaf)  removeFromLeaf (idx);
       else  removeFromNonLeaf (idx);
    }
```

```
else
{   if (leaf)
    {   cout << " does not exist\n";
        return;
    }
    bool flag = ((idx == n) ? true : false);
    if (child[idx]->n < 2)   fill(k);
    if (flag && idx >= n)
        child[idx-1]->remove(k);
    else
        child[idx]->remove(k);
}
return;
}
```

Auxillary Functions:

removeFromLeaf → shifts elements in left side after deletion

removeFromNonleaf → merges nodes after deletion

splitChild → splits a Node at child node

insertNonFull → inserts key into Node