2-3 trees writeup.

Pseudo code:

Insertion:

```
~~void insert(k)~~
void insert(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] -> insert NonFull(k);
            root = s;
        } else
            ~~root~~ root -> insert NonFull(k);
}
```

Deletion:

```
void delete(int k) {
    int idx = find(k);
    if (idx < n && keys[idx] == k) {
        if (leaf) remove from Leaf(idx);
    else remove NodeLeaf(idx);
}
else {
        if (leaf) {
            ~~froot~~ cout << "doesn't exist in";
            return ;
        }
    bool flag = ((idx ==n) ? true : false);
    if (child [idx] → n < t) fill (idx);
    if (flag && idx < n)
            child [idx - 1] → remove (k);
    else child [idx] → remove.
  }
    return ;
}
```

Auxillay function ;

remove from leaf → shift elements to left side after deletion

remove from non leaf → merge nodes after deletion

split child = splits a node into children node

insert non full → inserts key into node