

2-3 tree:

```
class treenode
```

```
{ int *keys;
```

```
  treenode **child;
```

```
  int n,
```

```
  bool leaf;
```

```
  friend class tree;
```

```
};
```

```
class tree
```

```
{ treenode *root = NULL;
```

```
  void traverse() {
```

```
    if (root != NULL)
```

```
      root->traverse();
```

```
  }
```

```
  void insert (int k);
```

```
  void delete (int k);
```

```
};
```

```
void tree::insert (int k)
```

```
{ if (root == NULL) { 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); }
```

```
}
```

```
void treenode:: insertnotfull(int k)
```

```
{
```

```
    int i = 0 - 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 (child[i+1] -> n == 3)
```

```
            {
```

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

```
                if (keys[i+1] < k) i++; }
```

```
                child[i+1] -> insertnotfull(k); }
```

```
    }
```

```
void treenode:: remove(int k)
```

```
{ int idx = findkey(k)
```

```
    if (idx < n && keys[idx] == k) { if (leaf) removefromleaf(idx);
```

```
        else removefromnonleaf(idx);
```

```
    } else if (leaf) { print "key doesn't exist"; return; }
```

```
    bool flag = (idx == n) ? true : false;
```

```
    if (child[idx] -> n < 2) flag;
```

```
    if (flag && idx > n) child[idx-1] -> remove(k);
```

```
    else child[idx] -> remove(k); }
```

```
    return; }
```