2-3 Tree

Nitish .N. Banakar
1BM18CS065

```
Class Tree{
  Tree * root;
   int t ; //degree
  Tree(){
     root = NULL;
      t=2;
  }
};
void insert (int k){
    if (!root){
        root = new TNode();
        root -> keys[0]= k;
        root->n = 1;
    }
    else if (root->n == 2*t-1){
      Node * s = new Node();
     s->~~s~~[0] -> root;
      s -> split (0, root);
      int i=0;
      if (s->keys[0]< k)  i++;
      s-> c[i] -> insertintoNode(k);
      root = s;
    }
    else
        root-> insertintoNode(k);
}
```

Nitish

Nitish.N.Banakar
1BM18CS068

```
void remove (int k){
    if (!root)  return;
    root -> remove (k);
    if (root -> n== 0) {
        Tree *  temp = root;
        if (root -> leaf) root = NULL
        else   root = root -> c[0];
        delete temp;
    }
    return;
}
void insertintoNode (int k){
    int i =n-1;
    if (leaf){
        while (i >=0  && key [i] > k) {
            keys [i+1] = keys[i];
            i--;
        }
    }keys[i+1] = k;
    n += 1;
    } else{
        while ( i >=0  && keys [i] > k)  i--;
        if (c[i+1] -> n == 2*t-1)
            split (i+1 ; c[i+1]);
        if (keys [i+1] < k)  i++;
    }
    c [i+1] -> insertintoNode (k);
}}
```

Nitish