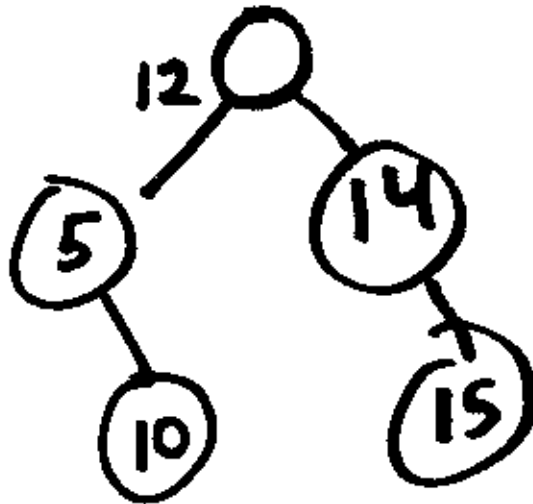
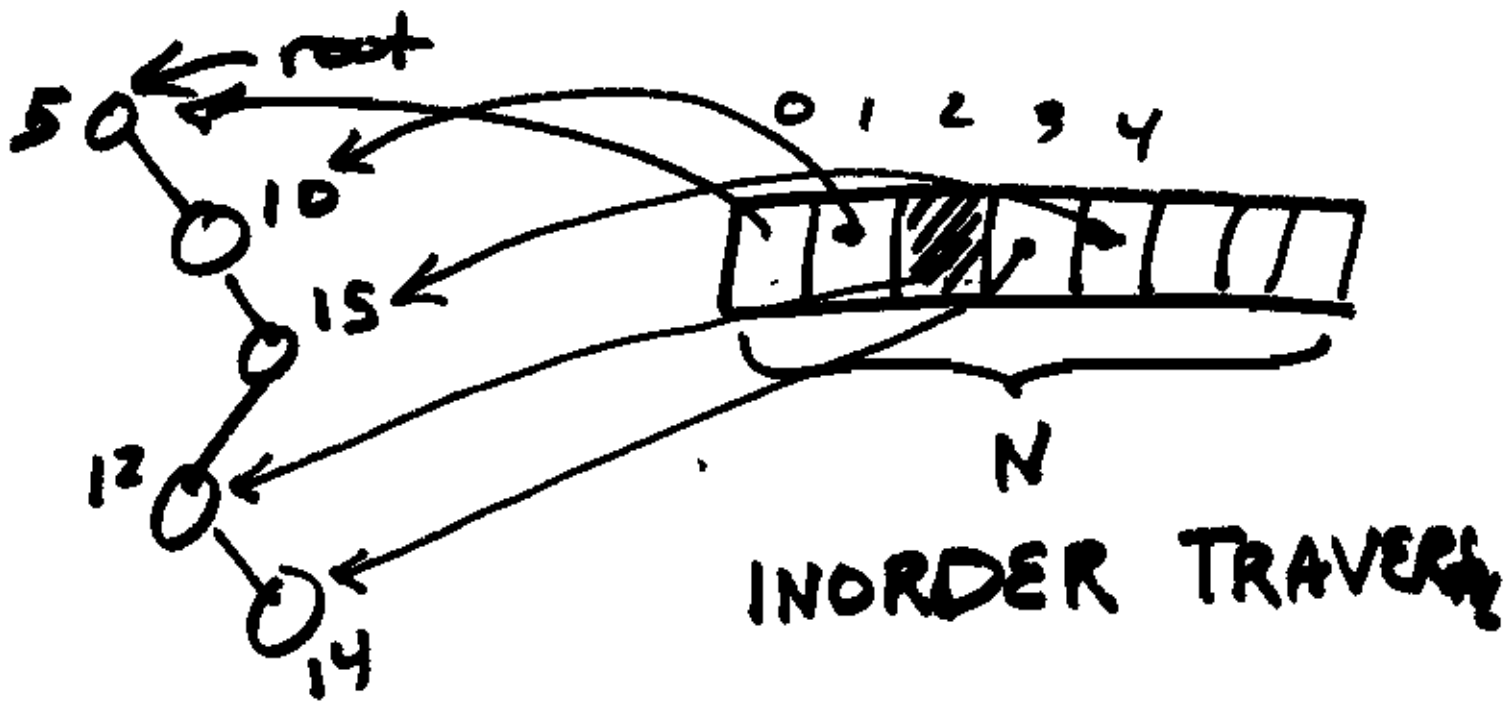


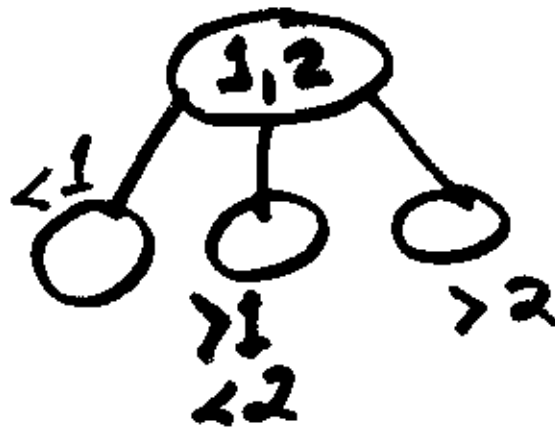
Balancing

- Brute Force (approach)
- 2-3 Tree } 100% Full
- 2-3-4 Tree } 100% Balanced
- Red Black Tree } Reasonably
- AVL tree } Balanced
- B-Tree

Brute Force



2-3 Tree



Struct node

```
{
  data * first;
  data * second;
  node * left;
  node * middle;
  node * right;
};
```

Struct node

```
{
  data * inside[2];
  node * child[3];
};
```

5 pointers

2-3 Tree Insert

- Travel down the tree using Binary Search technique, all the way to a leaf
- IF the leaf has only 1 piece of data, we add the data into the node.
- IF NOT, find the middle data item and split the Node.
"Push up" the middle.

3 5 40 15 20 5 45 3 50

Root



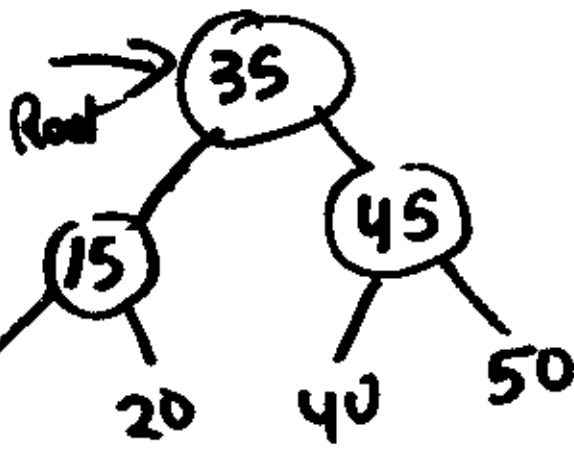
Root



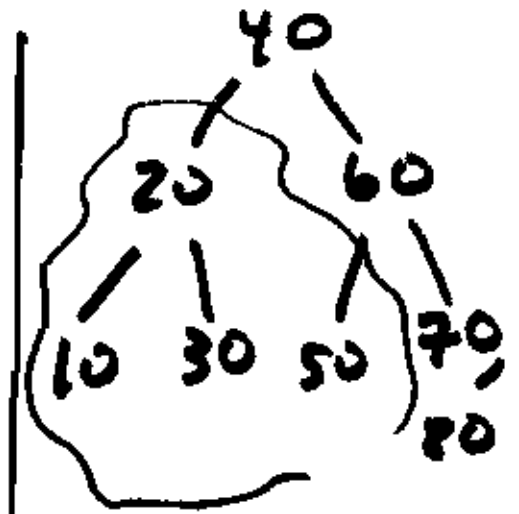
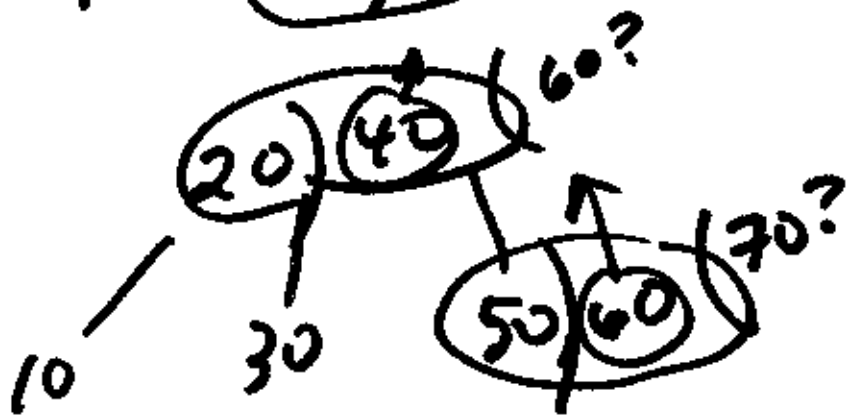
3, 5

20

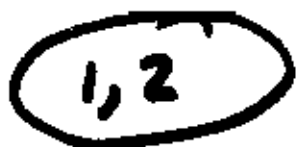
50?



10 20 30 40 50 60 70 80



2-3-4 Tree



Struct node

```
{ data * mydata [3];  
  node * child [4];
```

```
};
```

7 pointers