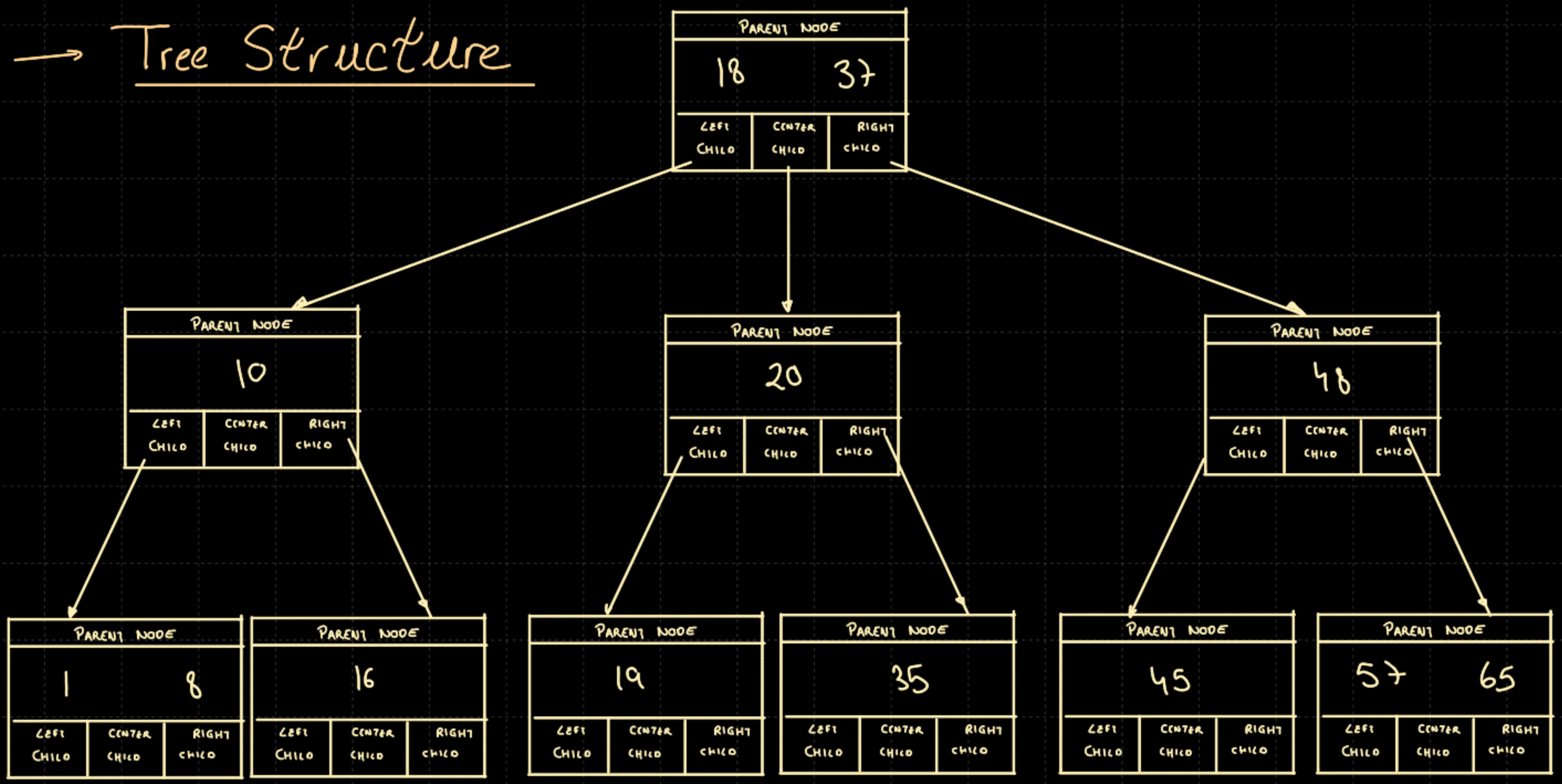
→ Tree Structure

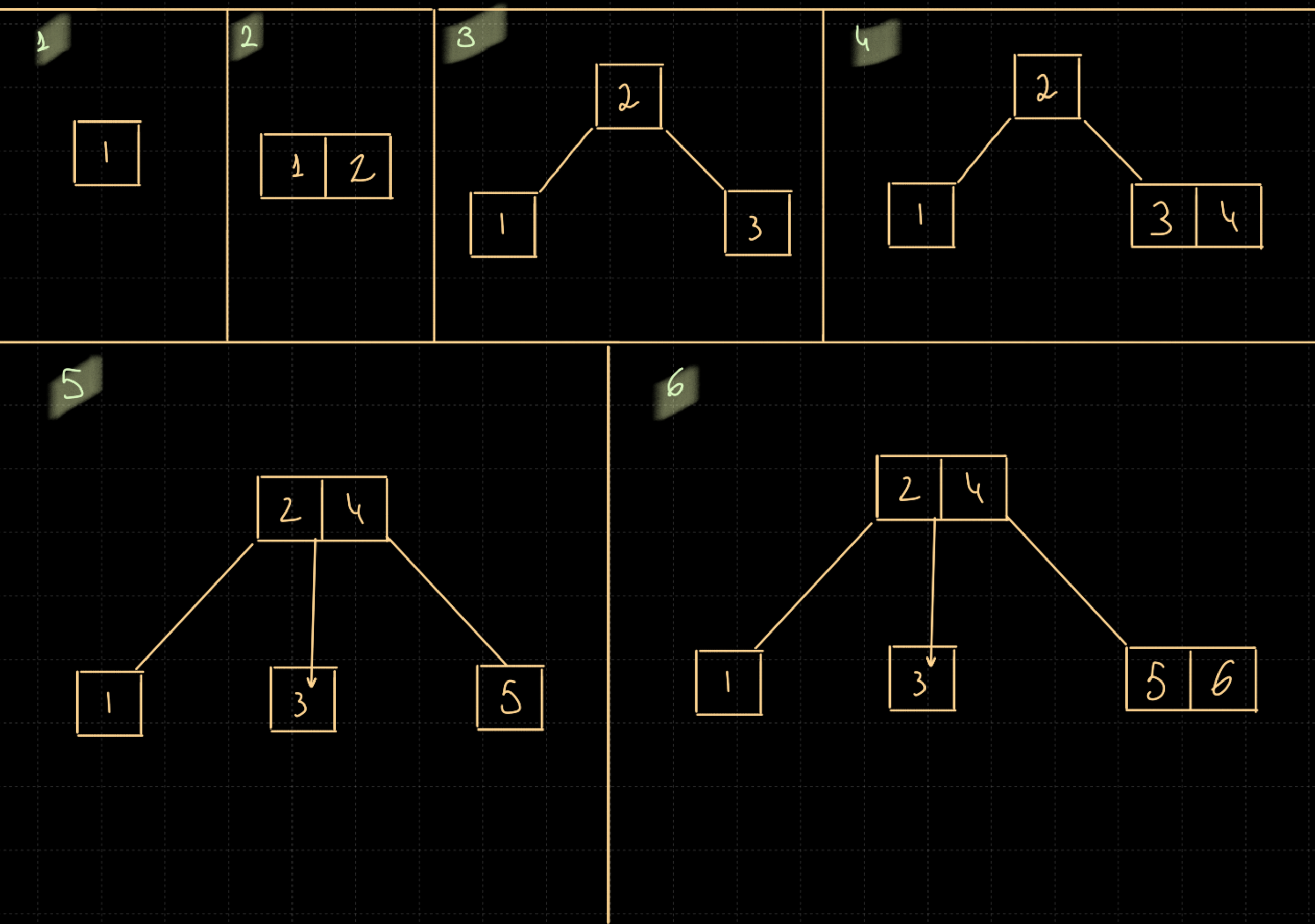


Attributes → root Node

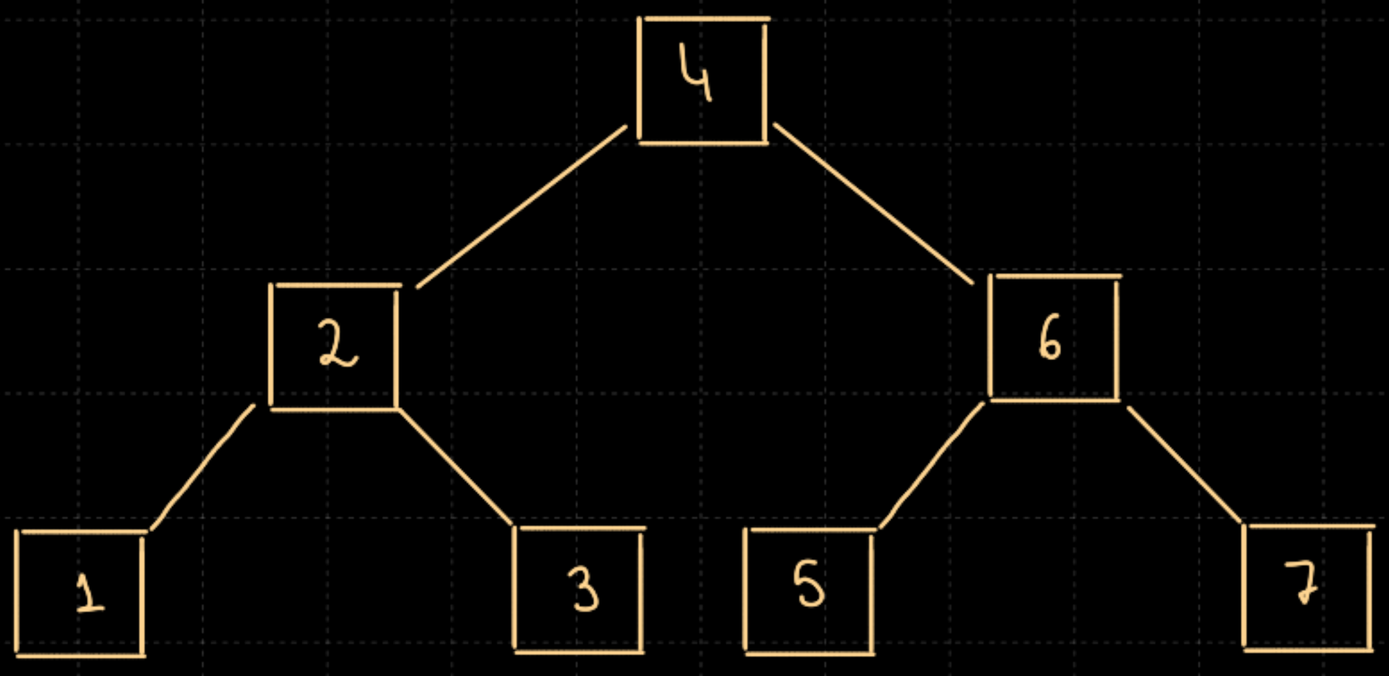
→ degree of tree = 3

1. Insertion Algorethm

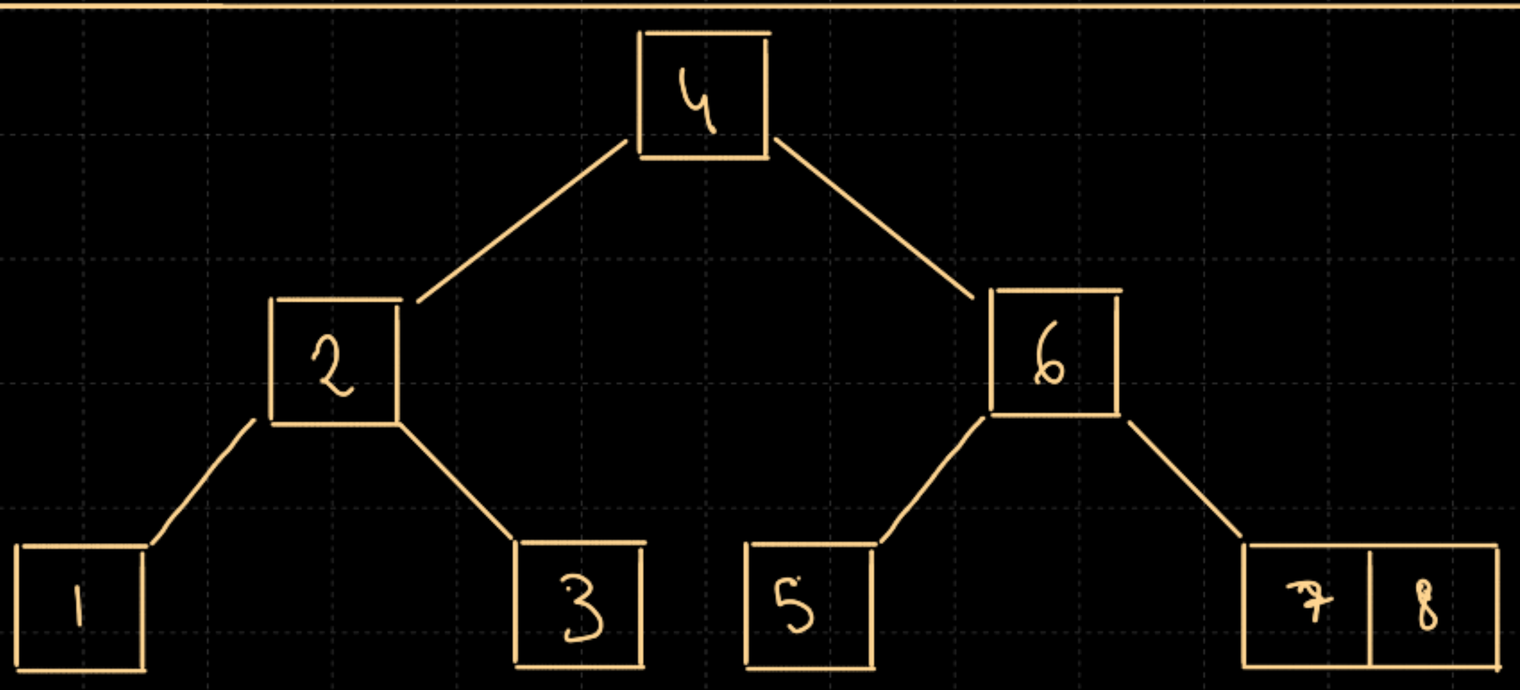
How would the tree look like after the insertion of the first 15 elements ?



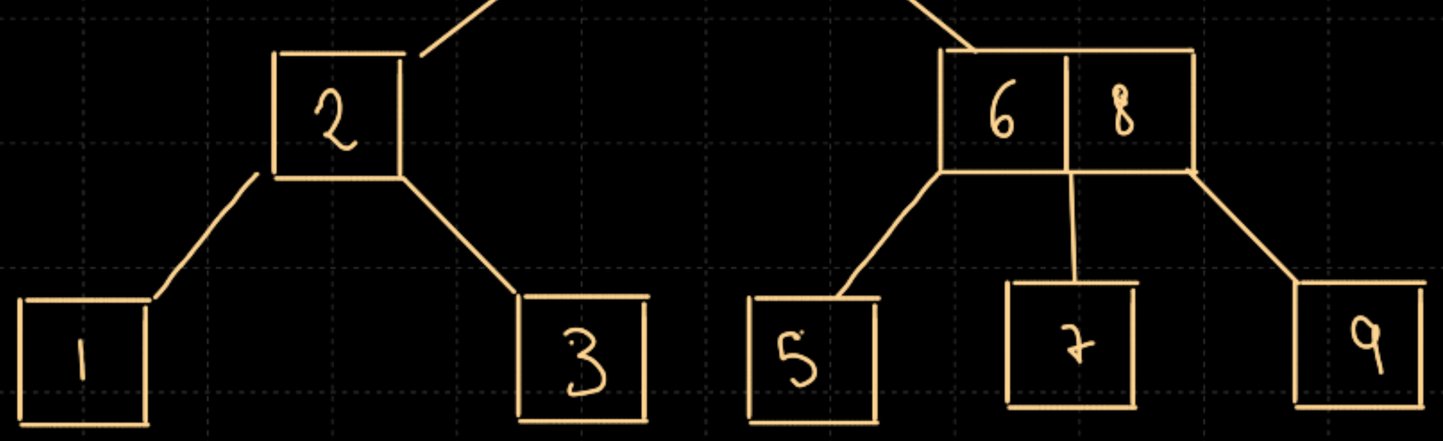
7



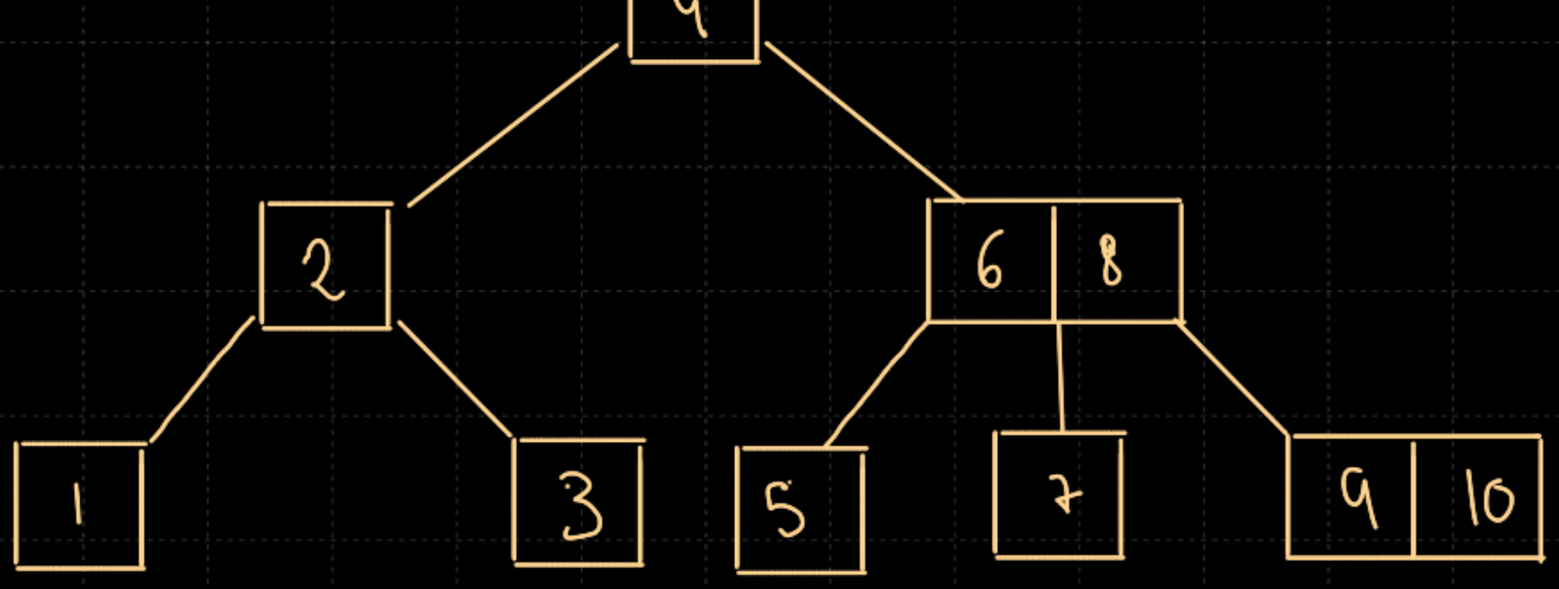
8



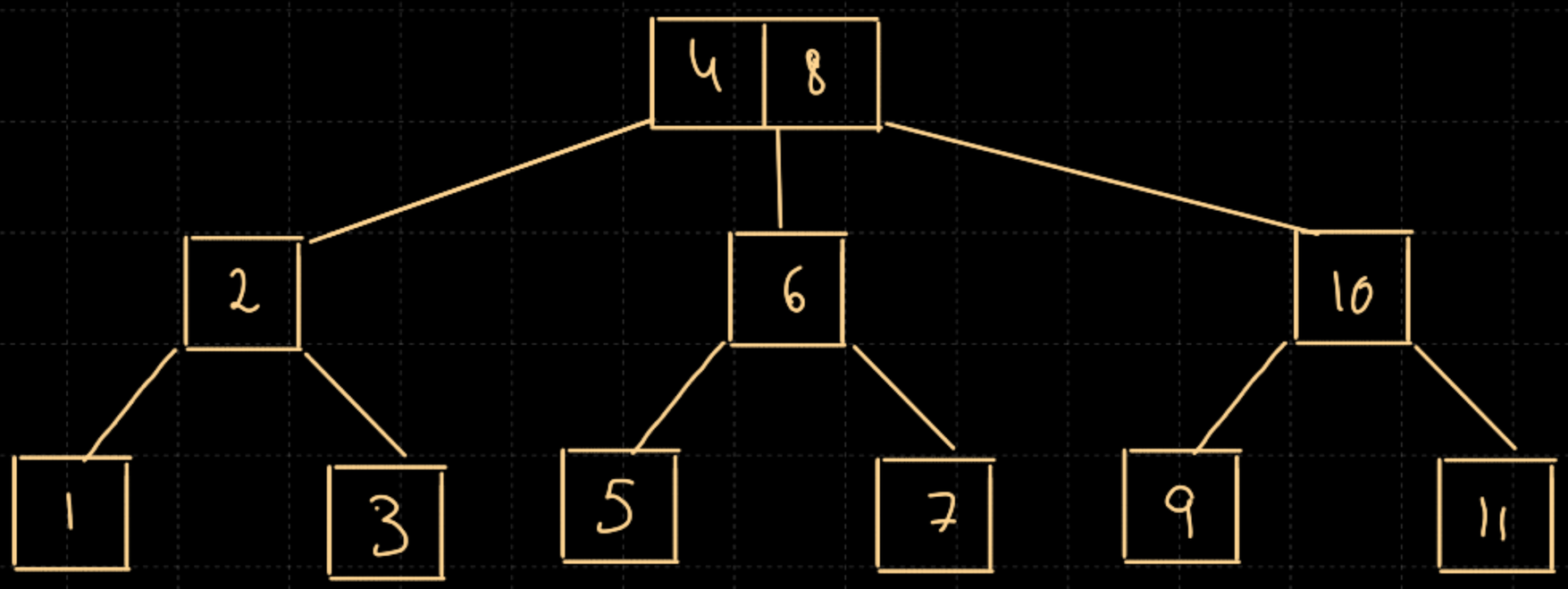
9



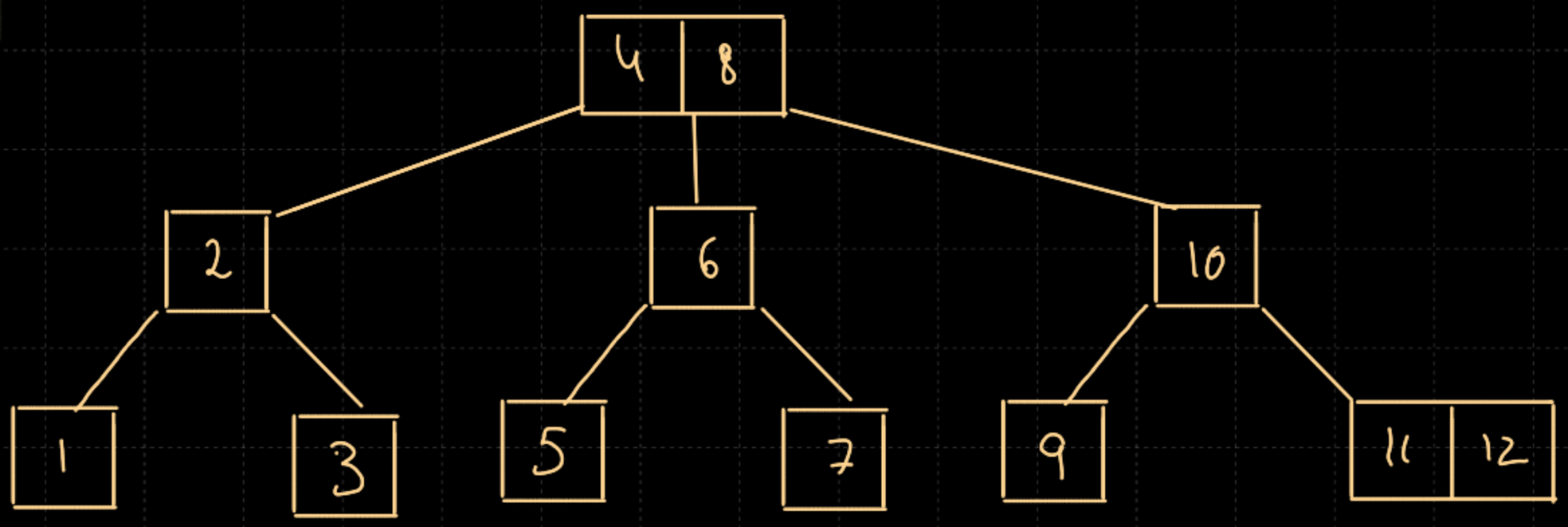
10



11


4
8
2
6
10
1
3
5
7
9
11

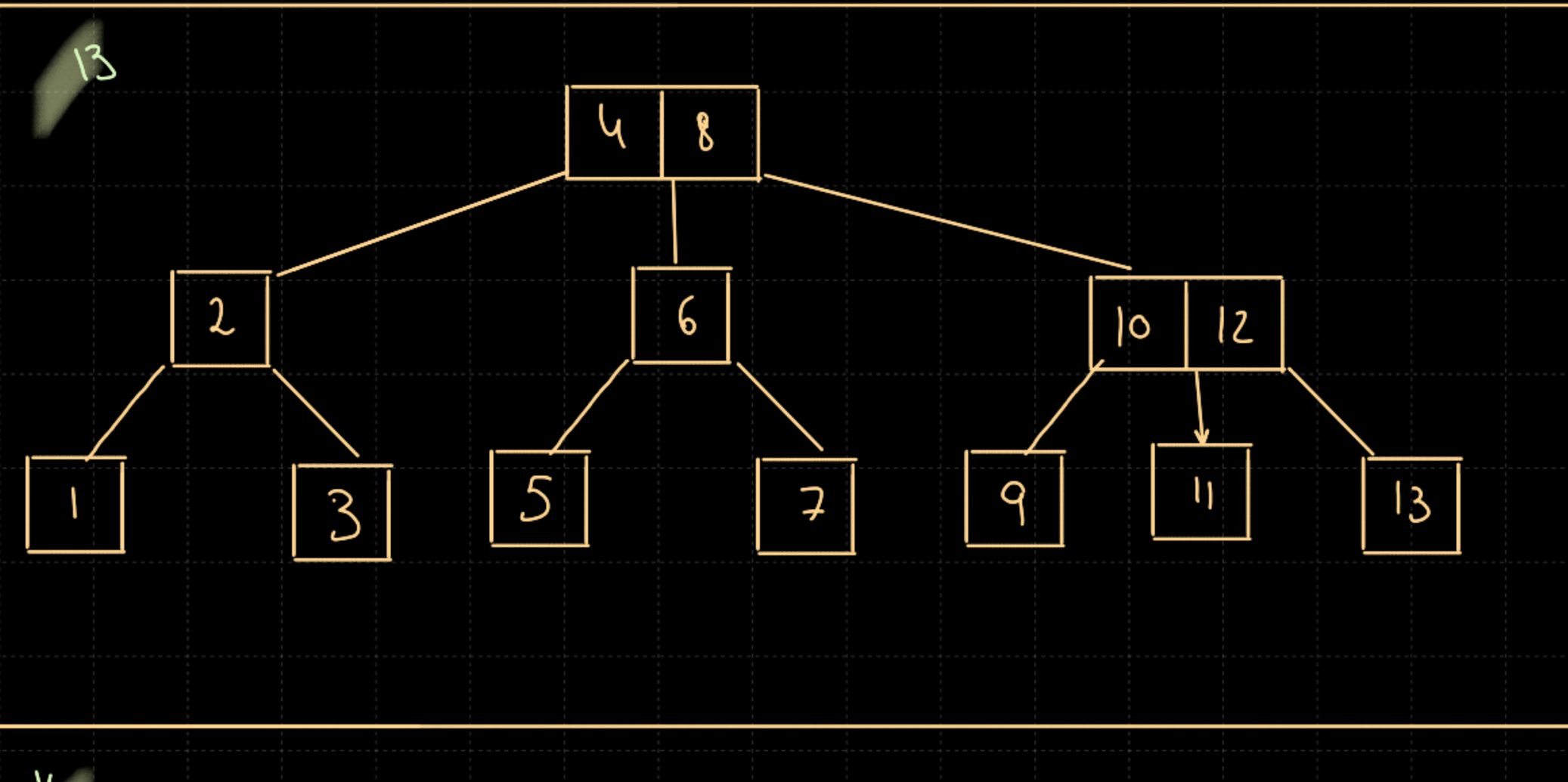
12

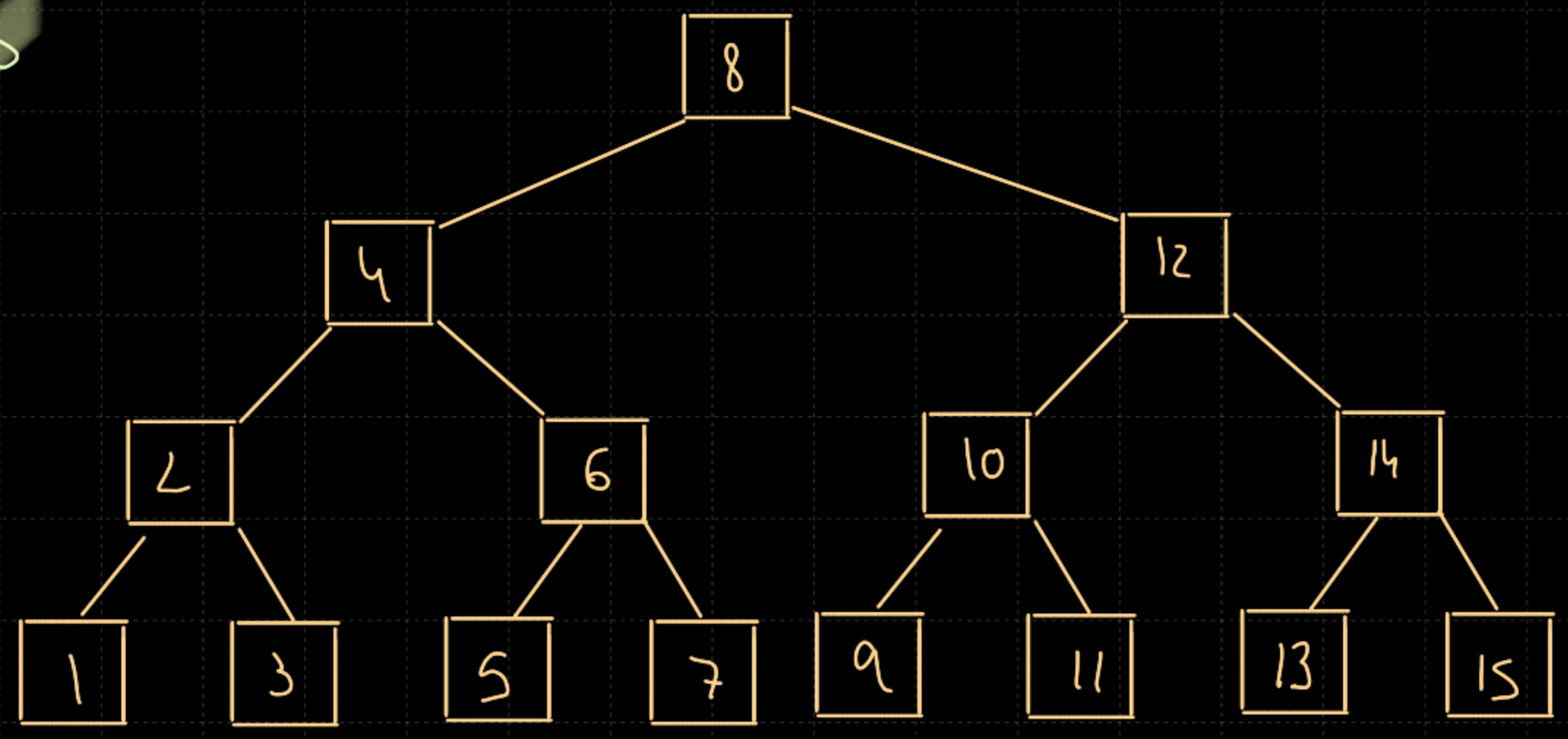

4
8
2
6
10
1
3
5
7
9
11
12

13

4
8
2
6
10
12
1
3
5
7
9
11
13

14


4
8
2
6
10
12
1
3
5
7
9
11
13
14



Insertion Algorithm (Value)

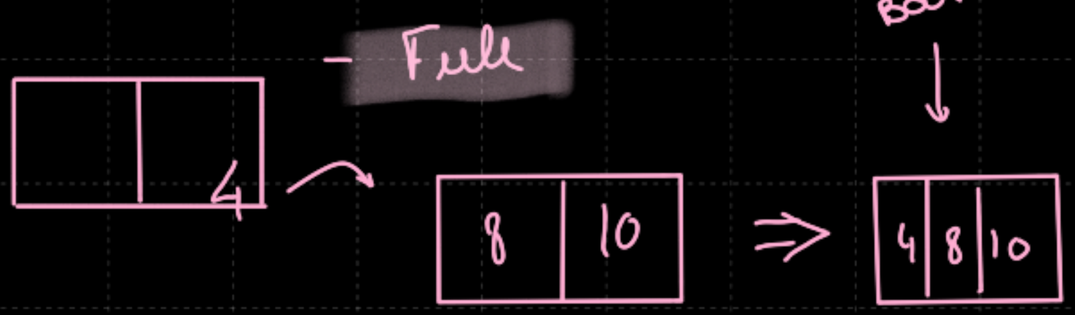
```
r ← root
If r.Size == Upper Bound
    S ← new Node
    root ← S
    S.leaf ← false
    S.size ← 0
    S.children(0) ← r
    SplitChild (parent, index, child)
    InsertNonFull ( S, value )
else
    InsertNonFull ( r, value)
end
end
```

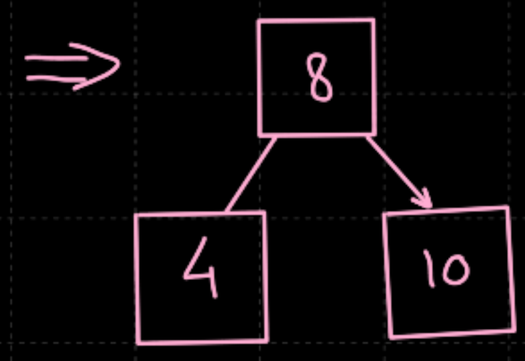
2 ← Max number of keys per node

Casos de inserción :

a) Root :

- Full





- No full :

InsertNonFullAlgorithm (Key , Node)

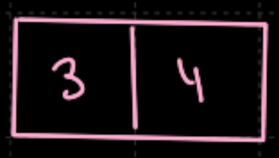
```
Index ← key[].size
If node is leaf
    while Key > K.[i] && index >=0)
        index ← index - 1
    end
    Insert value in index+1 in Keys[]
    if node size == UPPBOUND
        delete last Key
    end
    node.Size ← nde.size +1
else
    while Key > K.[i] && index >=0)
        index ← index - 1
    end
    index ← index +1
```

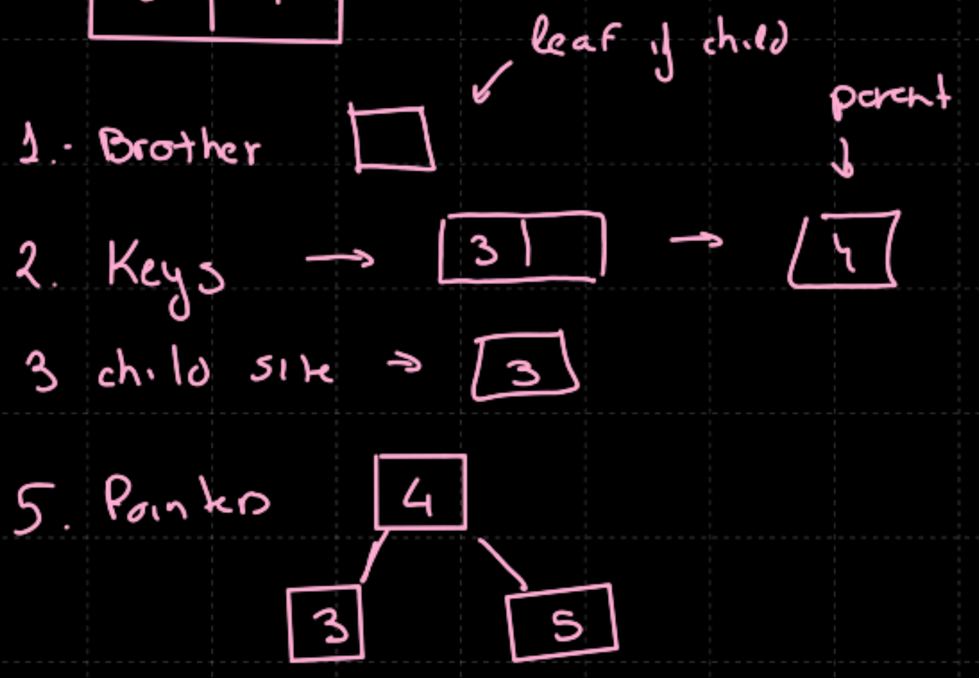
```
        if child[i].size == UPPBOUND
            Split Node (node, index, parent)
            if Key > node K[index]
                i ← i + 1
            end
        end
        insertNonNull (child[i], Key)
    end
end
```

SplitChild (parent, index, child)

```
    brother ← new Node
    brother.leaf ← child.isLeaf
    brother.size ← UPPER - LOWER - 1

    for j from 0 until new_Node.size - 1
    |   brother.K[] ← child.K[j + LOWER + 1]
    end

    if !child.isLeaf
    |   for j=0 untill brother.size
    |   |   brother.child[j] ← child.child[j + LOWER + 1]
    |   end
    end

    child.size ← LOWER BOUND
    parent.child[i+1] ← brother
    if padre.size == UPPER
    |       remove last key
    end

    parent.K[index] ← child.Keys[LOWER]
    if padre.size == UPPER
    |   Remove last key
    end

    parent.site ← parent.site + 1
end
```

5 ↷





4. In case had childs