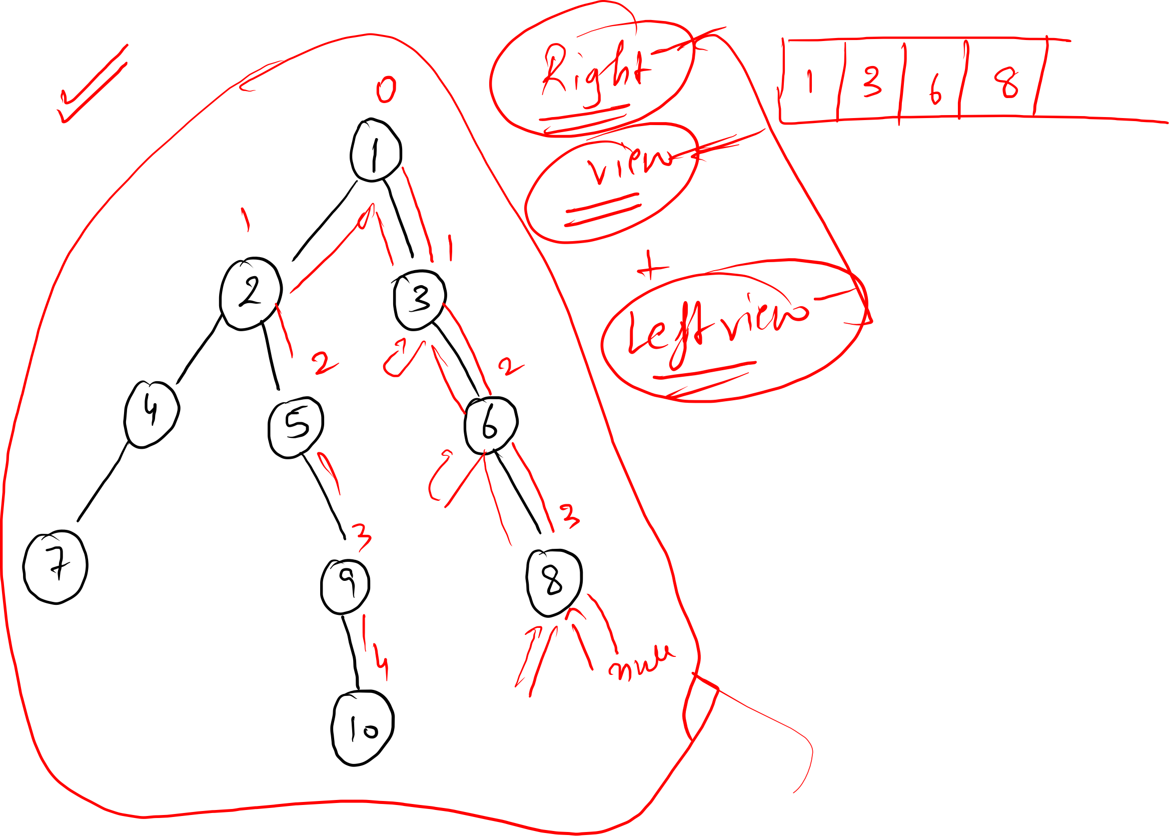



```java
class Solution {
    List<Integer> rightside = new ArrayList();

    public void helper(TreeNode node, int level) {
        if (level == rightside.size())
            rightside.add(node.val);

        if (node.right != null)
            helper(node.right, level + 1);
        if (node.left != null)
            helper(node.left, level + 1);
    }

    public List<Integer> rightSideView(TreeNode root) {
        if (root == null) return rightside;

        helper(root, 0);
        return rightside;
    }
}
```

WAR-MINES

Min Moves

Recursion

I

DP

0 → unsafe due to mine

1 → safe

2 → unsafe due to proximity to mine

3 → visited

Src | Dest

	0	1	2	3	4	5	6	7	8	9
0	1	1	1	1	1	1	1	1	1	1
1	1	0	1	1	1	1	1	1	1	1
2	1	1	1	0	1	1	1	1	1	1
3	1	1	1	1	0	1	1	1	1	1
4	1	1	1	1	1	1	1	1	1	1
5	1	1	1	1	1	0	1	1	1	1
6	1	0	1	1	1	1	1	1	0	1
7	1	1	1	1	1	1	1	1	1	1
8	1	1	1	1	1	1	1	1	1	1
9	0	1	1	1	1	0	1	1	1	1
10	1	1	1	1	1	1	1	1	1	1
11	1	1	1	0	1	1	1	1	1	1

13

i 1 i 1 i 1 i

3

(0,0) → +∞
(2,0) → 16
(3,0) → 15
(4,0) → 14
(5,0) → 15
(6,0) → +∞
(7,0) → +∞
(8,0) → +∞
(9,0) → +∞
(10,0) → +
(11,0) → 1

```
0,0 -> 2147483647
1,0 -> 2147483647
2,0 -> 16
3,0 -> 15
4,0 -> 14
5,0 -> 15
6,0 -> 2147483647
7,0 -> 2147483647
8,0 -> 2147483647
9,0 -> 2147483647
10,0 -> 2147483647
11,0 -> 15
```

{5, 4, 3, 2}

True → < , false → >

arr[i-1] < arr[i]

↳ True ✓

↳ false → swap

arr[i-1] > arr[i]

↳ T ✓

↳ false → swap

a < b > c < d > e < f.

0	1	2	3
4	5	2	3

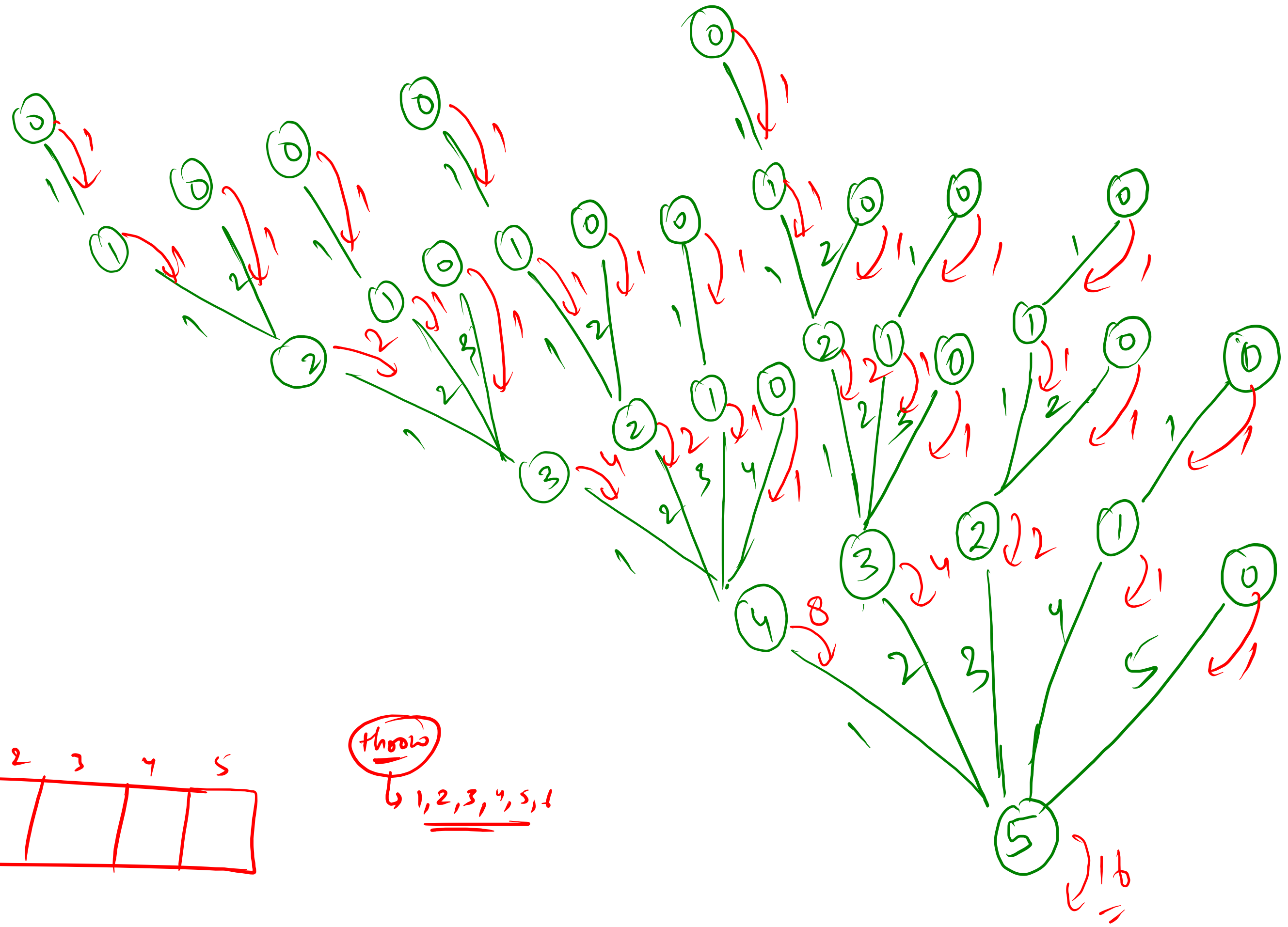
!flag = true

```
for (idx = 1; idx < len; idx++){
    if (flag) {

    }else {

    }
    flag = !flag;
}
```

Count

print

AList



5 N
6 m

0	1	2	3	4	5
src					

throw

1, 2, 3, 4, 5, 6

L-I → TREES → Thursday → 29/07/21 → Time: 8:00 – 10:00

A group of friends(divyanshu,abhishek,mayank,raman) are playing a game nd they call it "quarantine game".
game is decribed as followed :
1. a person is given a string of characters.
2. in order to escape from quarantine ,that person has to find total-count of very special kind substring.
3. only substring having same first nd last character are selected in the total-count.

Now, divyanshu is selected at random & locked in a dark room.he is given a string, your task is to help him
find correct number of special substrings so that he can get out from the room.

Substrings

$O(n)$, $O(n \log n)$

$O(n^2)$

abcdcbab

$a \to 3$
$b \to 3$
$c \to 2$
$d \to 1$

$^nC_2$

ch → Sing
substring (2 select) $^nC_2$

a
ab
abc
abcd
abcdc
abcdcb
abcdcba

b
bc
bcd
bcdc
bcdcb
bcdcba

c
cd
cdc
cdcb
cdcba

d
dc
dcb
dcba

c
cb
cba

b
ba

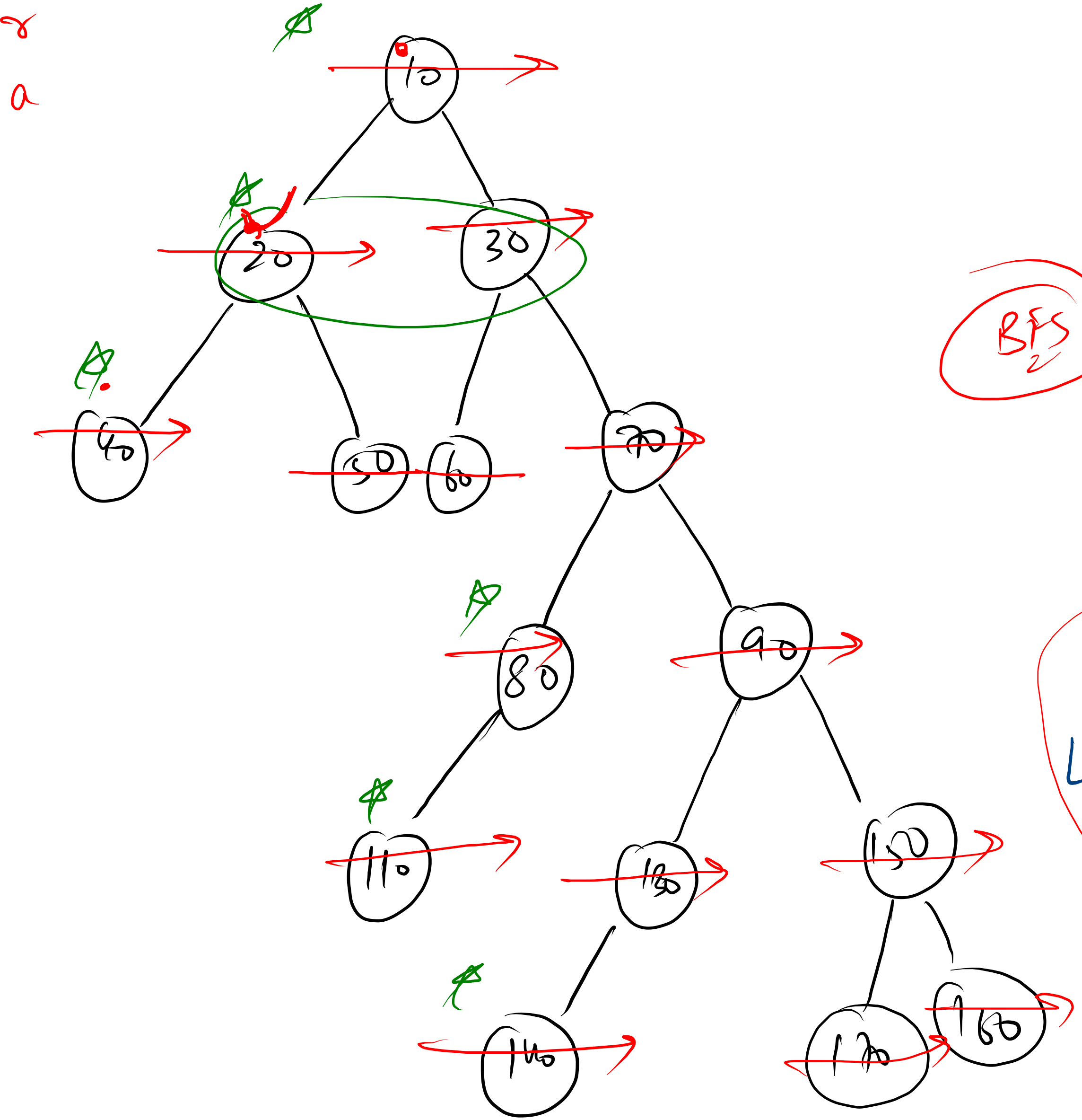
a

Tree

left view
right view
Top view
Bottom view

vertical order

Reverse Level Order



BFS

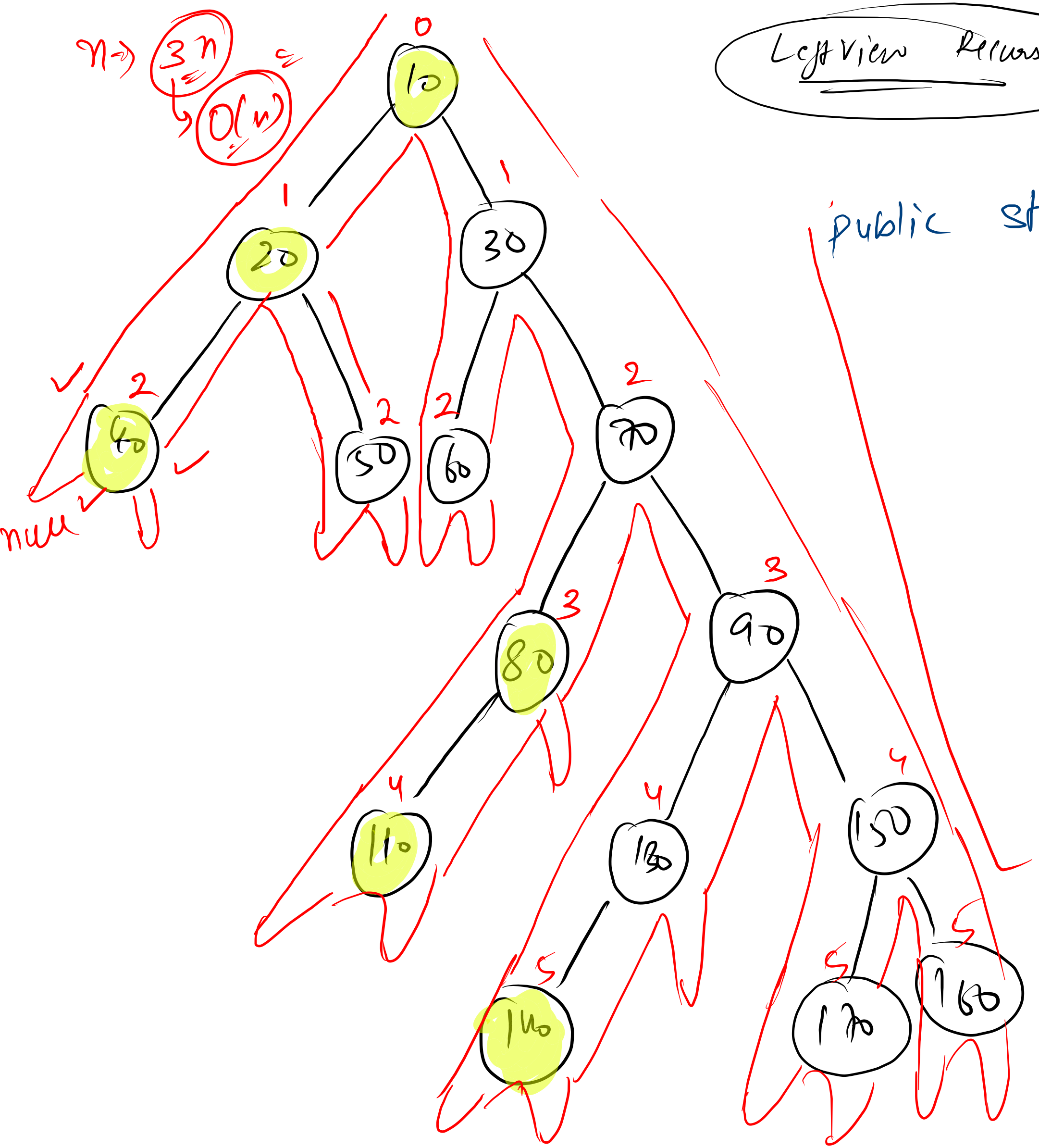
currL = 4u

nextL = 5u

LV:

10	20	40	80	110	140

Left View Recursive



list = | 10 | 20 | 40 | 80 | 110 | 140 |

DFS

```
public static void LeftView (Node node, int level, ArrayList<Integer> list){
    if(node == null) { return; }

    if(level == list.size()){
        list.add(node.data);
    }

    leftView(node.left, level+1, list);
    leftview(node.right, level+1, list);
}
```

T.C. -> O(n)

S.C. -> O(h)

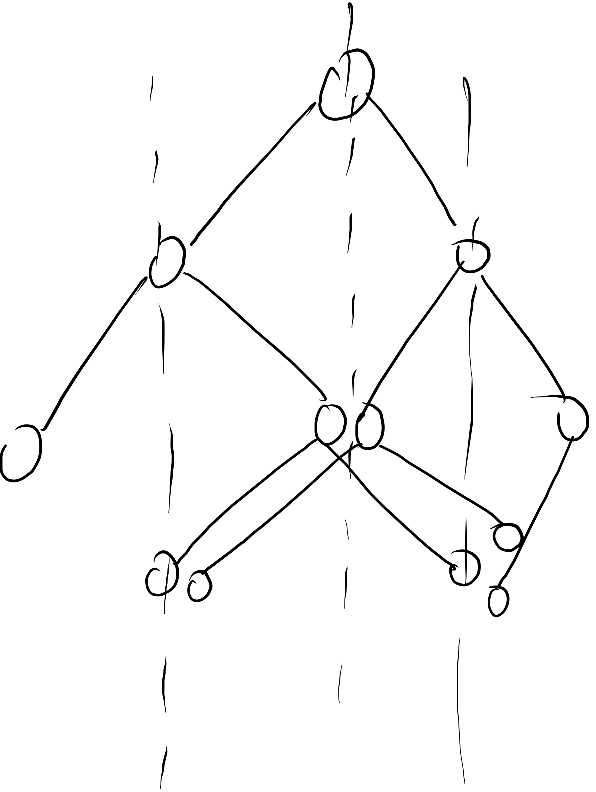
height -> n

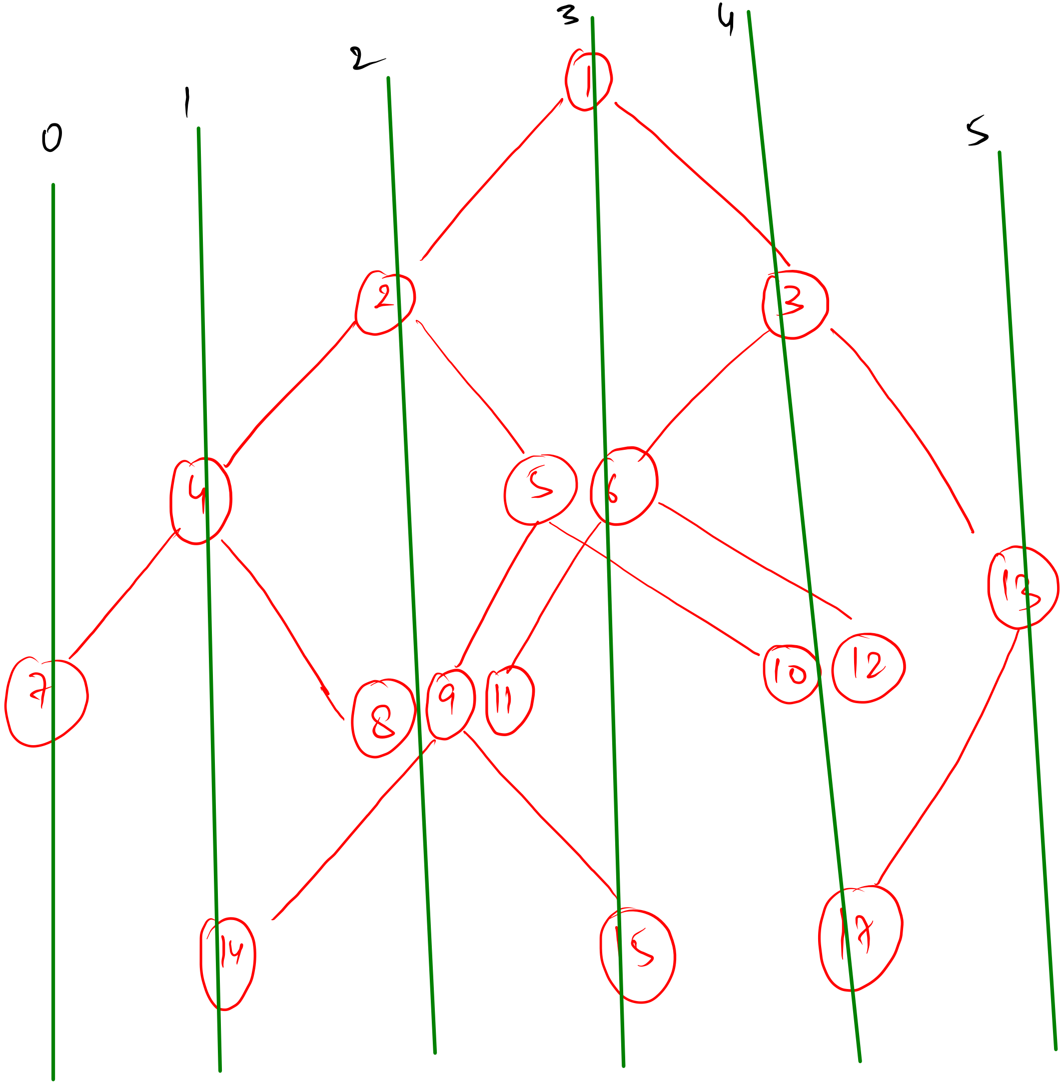
Left View

Right View

BFS, DFS

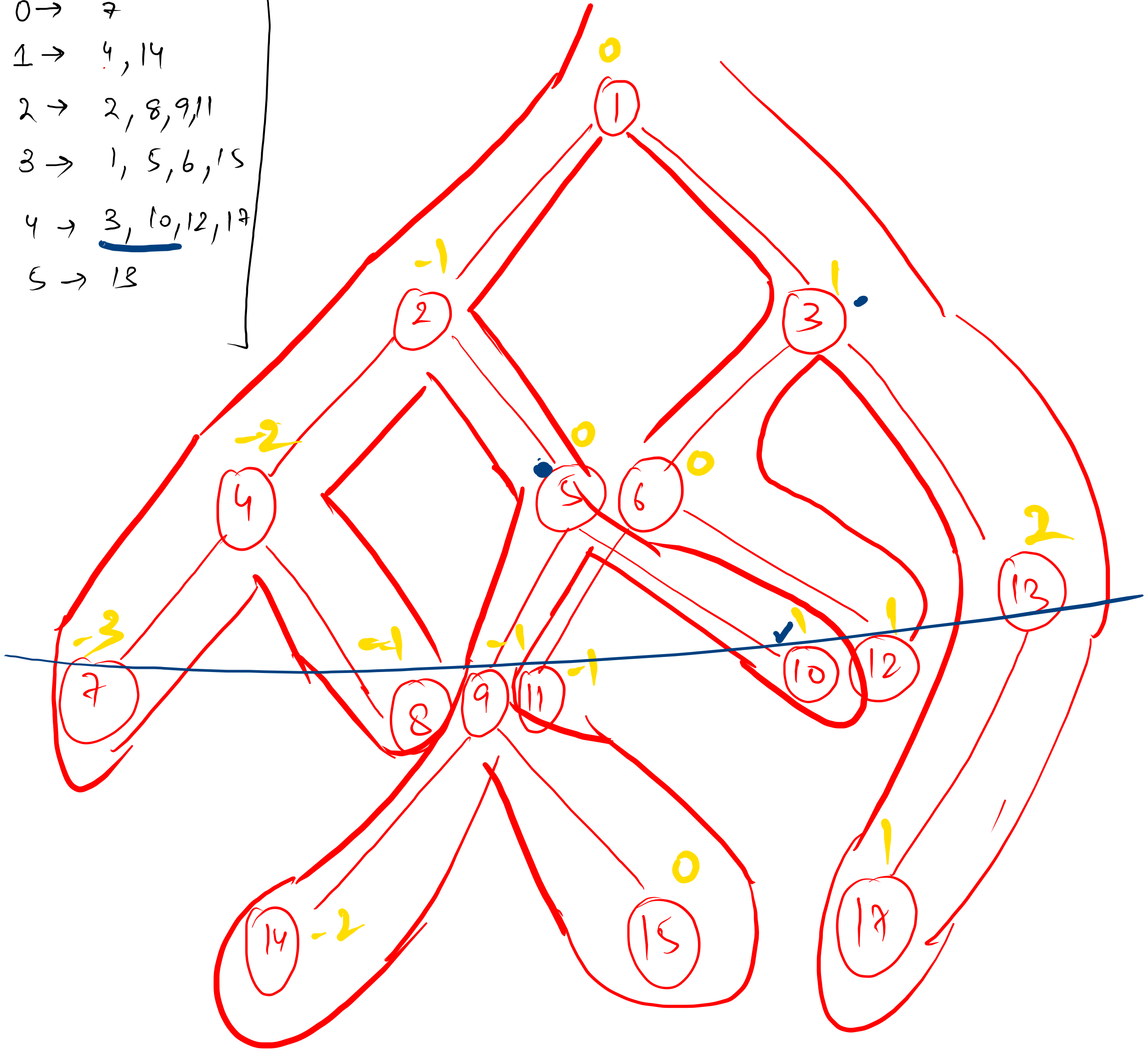
Vertical order





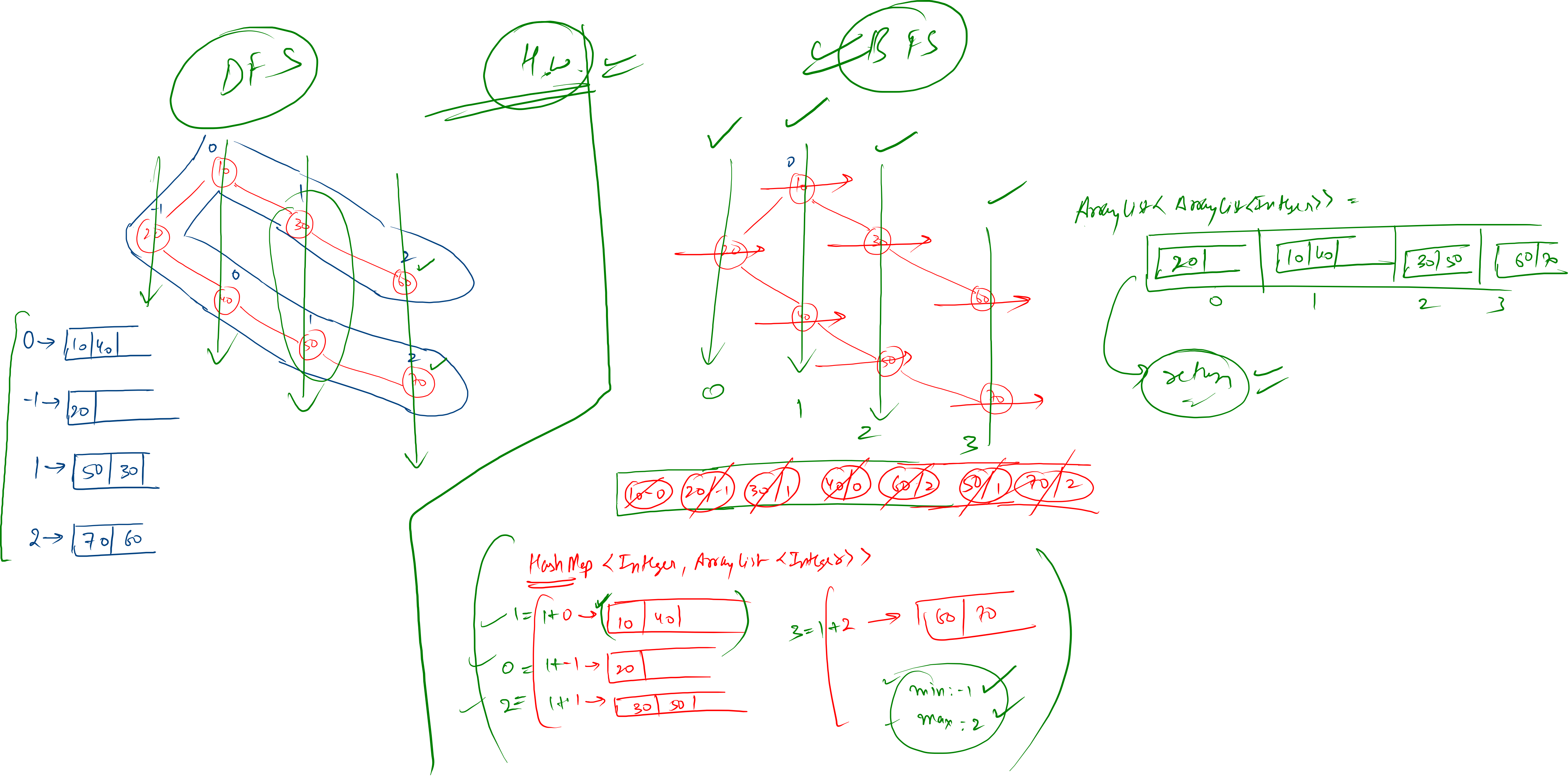
0 → 7
1 → 4, 14
2 → 2, 8, 9, 11
3 → 1, 5, 6, 15
4 → 3, 10, 12, 17
5 → 13

0 →	7	
1 →	4, 14	
2 →	2, 8, 9, 11	
3 →	1, 5, 6, 15	
4 →	3, 10, 12, 17	
5 →	13	



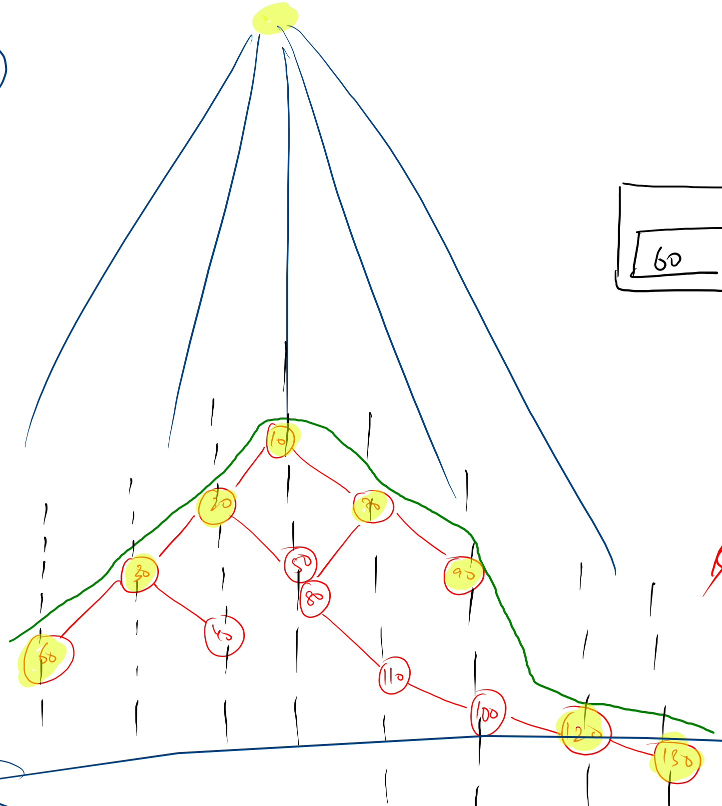
HashMap<Integer, ArrayList>

0 →	1 | 5 | 15 | 6	✗
-1 →	2 | 8 | 9 | 11	✓
-2 →	4 | 14	✓
-3 →	7	✓
1 →	10 | 3 | 12 | 17	✗
2 →	13	✓



DFS
H.W
BFS
0 → 10 | 40
-1 → 20
1 → 50 | 30
2 → 70 | 60
ArrayList<ArrayList<Integer>> =
20
10 | 40
30 | 50
60 | 70
0
1
2
3
return
10/0
20/-1
30/1
40/0
60/2
50/1
70/2
HashMap <Integer, ArrayList <Integer>>
1 = 1+0 → 10 | 40
0 = 1+-1 → 20
2 = 1+1 → 30 | 50
3 = 1+2 → 60 | 70
min : -1
max : 2

Top view



60, 30, 20, 10, 70, 90, 120, 130

60	30	20 | 40	10 | 50 | 80	90 | 100	120	130

Left View → Levelorder, Rec
Right View → = , =
V.O. → BFS, DFS
Top → V.O.
Bottom → V.O.

first element of each vertical order is

Top view

60	30	20	10	90	120	130

Bottom View → last element of every vertical order is bottom view

H.W.

2 hours

</>	Left View Of A Binarytree	Easy	10	Auth	0	Public	Sol	9
</>	Right View Of A Binarytree	Easy	10	Auth	0	Public	Sol	10
</>	Vertical Order Traversal Of A Binarytree	Easy	10	Auth	0	Public	Sol	11
</>	Bottom View Of A Binarytree	Easy	10	Auth	0	Public	Sol	12
</>	Top View Of A Binarytree	Easy	10	Auth	0	Public	Sol	13

12 Target

```
List<Integer> list = new ArrayList<Integer>();
```