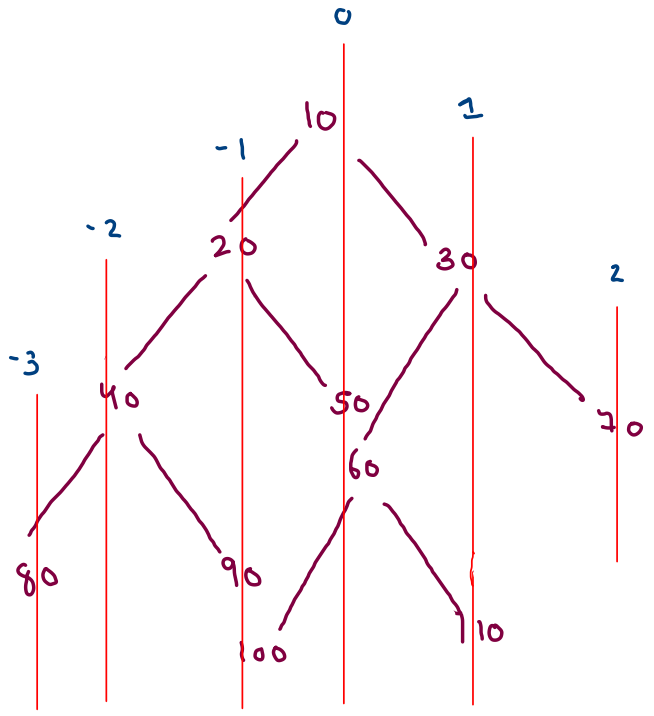
Width Of Shadow Of Binary Tree



width = total vertical lines

vot

80

40

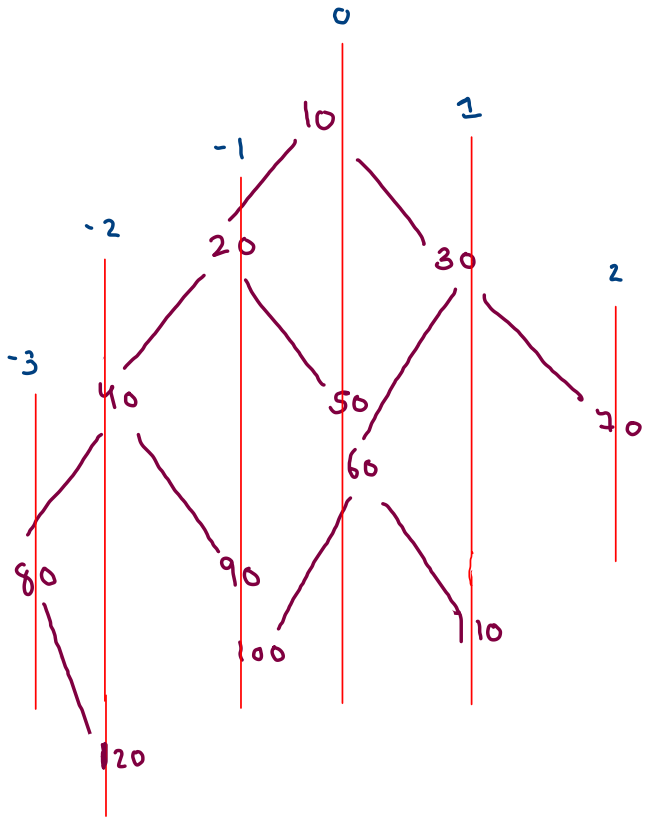
20 90 100

10 50 60

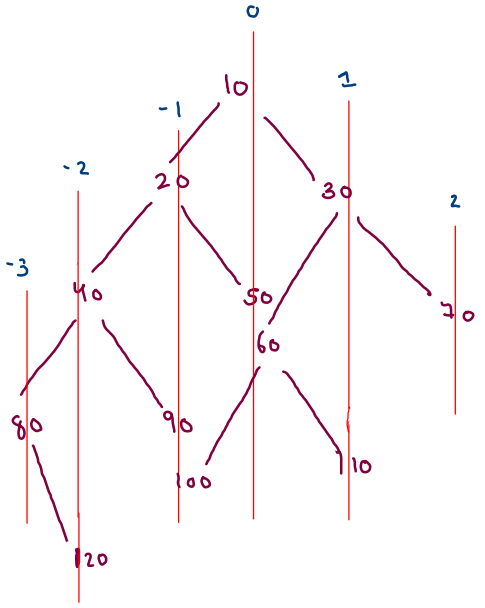
30 110

70

Vertical Order Traversal Of A Binarytree



Vl	nodes (T to B)
-3	80
-2	40 120
-1	20 90 100
0	10 50 60
1	30 110
2	70



map :

key : vtn or X

vertical line no.

value : Al < Integer>

0 → 10, 50, 60

-1 → 20, 90, 100

1 → 30, 110

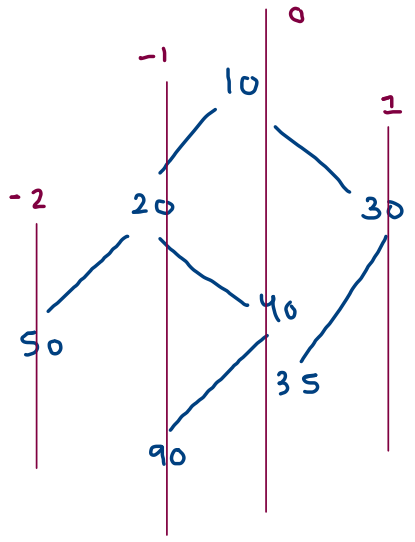
-2 → 40, 120

2 → 70

-3 → 80

10,0	20,-1	30,1	40,-2	50,0	60,0	70,2	80,-3	90,-1	100,-1	110,1	120,-2

```
while(q.size() > 0) {
    //remove
    Pair rem = q.remove();

    //work
    ArrayList<Integer>list = map.getOrDefault(rem.x,new ArrayList<>());
    list.add(rem.node.val);
    map.put(rem.x,list);

    //add children
    if(rem.node.left != null) {
        q.add(new Pair(rem.node.left,rem.x - 1));
    }
    if(rem.node.right != null) {
        q.add(new Pair(rem.node.right,rem.x + 1));
    }
}
```



sx = -2

dx = 1

vdn = -2 -1 0 1

10,0 | 20,-1 | 30,1 | 50,-2 | 40,0 | 35,0 | 90,-1

50

20, 90

10, 40, 35

30

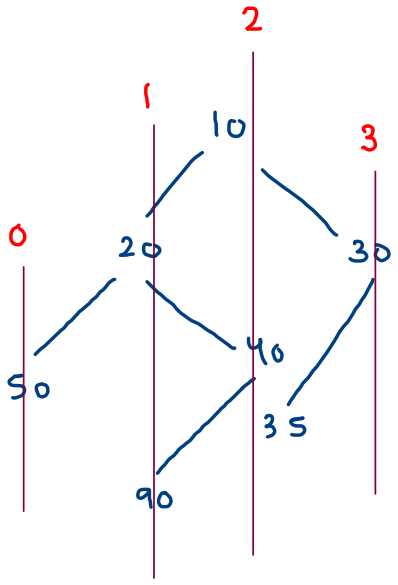
0 → 10, 40, 35

-1 → 20, 90

1 → 30

-2 → 50

VOT without map.

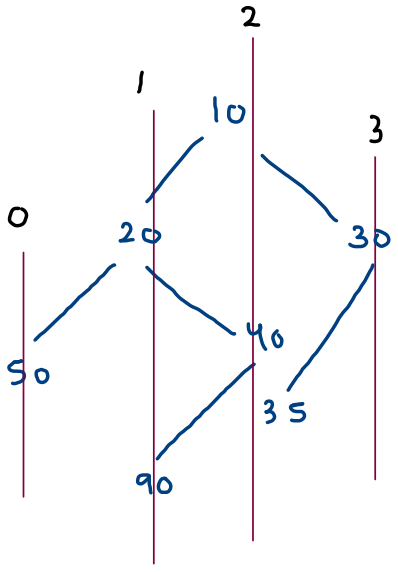


$sx = -2$ $dx = 1$ $w = 4$

root vln (x) = $-sx$

50	20, 90	10, 40, 35	30
0	1	2	3

10/2	20/1	30/3	50/0	40/2	35/2	90/1



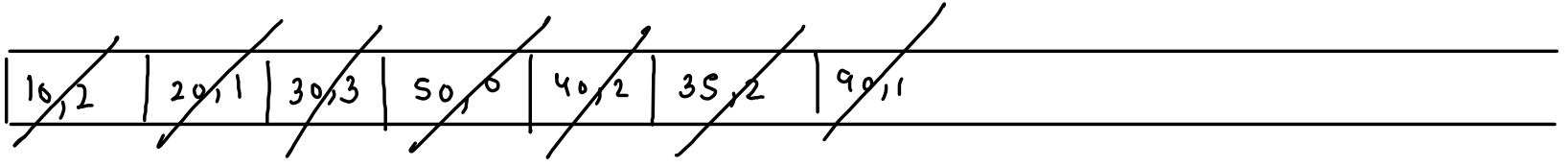

```
while(q.size() > 0) {
    //remove
    Pair rem = q.remove();

    //work
    ans.get(rem.x).add(rem.node.val);

    //add children
    if(rem.node.left != null) {
        q.add(new Pair(rem.node.left,rem.x-1));
    }
    if(rem.node.right != null) {
        q.add(new Pair(rem.node.right,rem.x+1));
    }
}
```

sx : -2

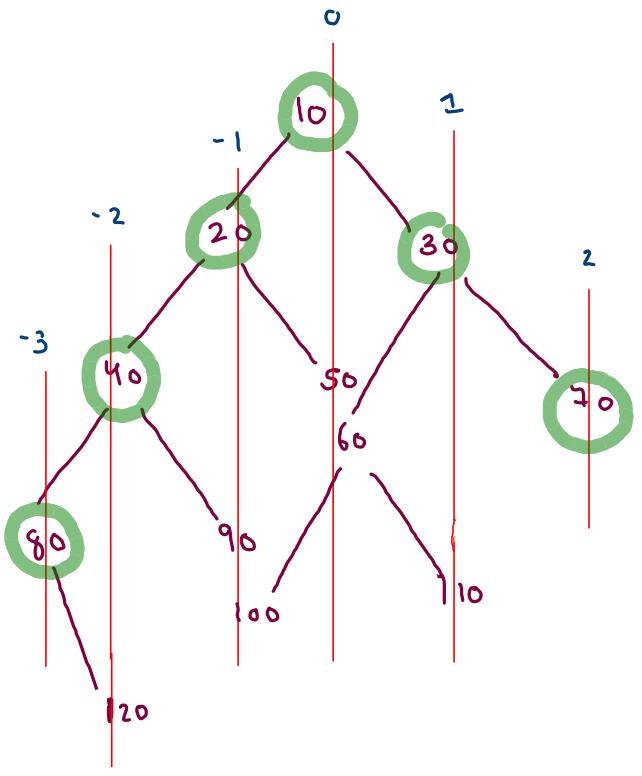
dx : 1

w : 4

50	20 , 90	10 , 40, 35	30
0	1	2	3

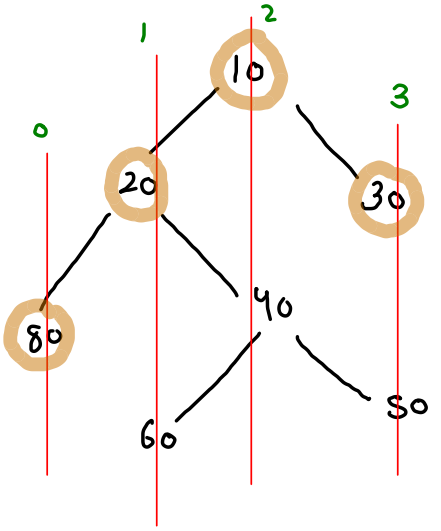


Top view



top view : first node of each vertical line

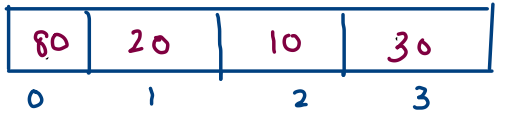
```java
while(q.size() > 0) {
    //remove
    Pair rem = q.remove();


    //work
    if(tv.get(rem.x) == -1) {
        tv.set(rem.x,rem.node.data);
    }

    //add children
    if(rem.node.left != null) {
        q.add(new Pair(rem.node.left,rem.x-1));
    }
    if(rem.node.right != null) {
        q.add(new Pair(rem.node.right,rem.x+1));
    }
}
```

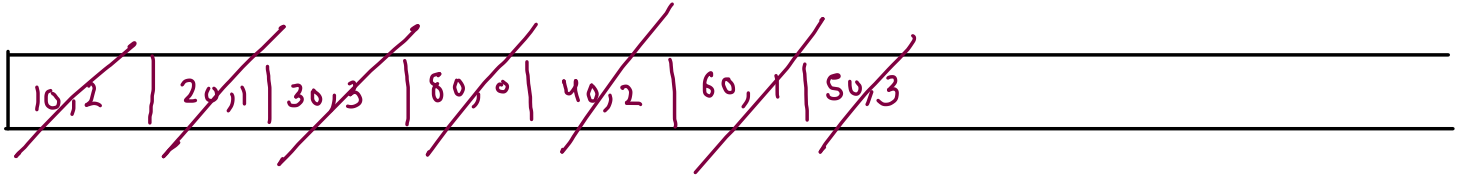


sx = -2

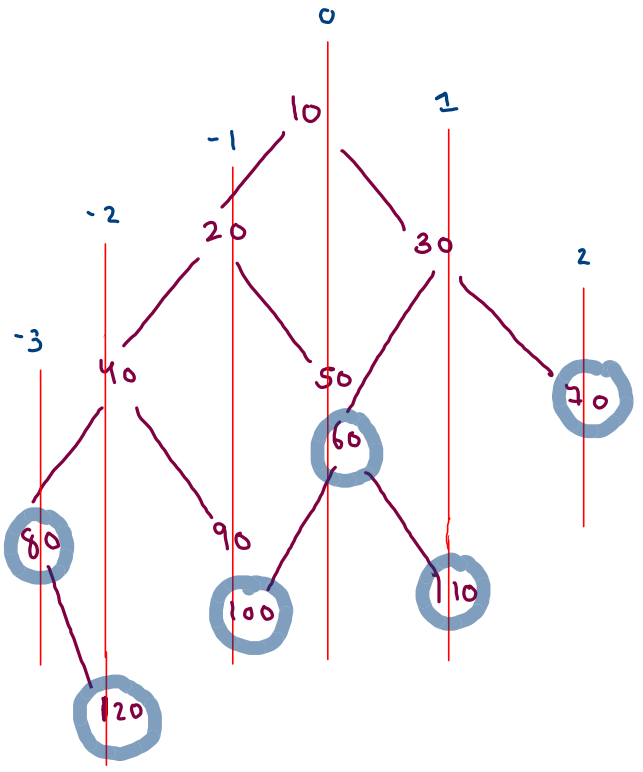
dx = 1

w = 4





Bottom view :



bottom view : last node of each vertical line.