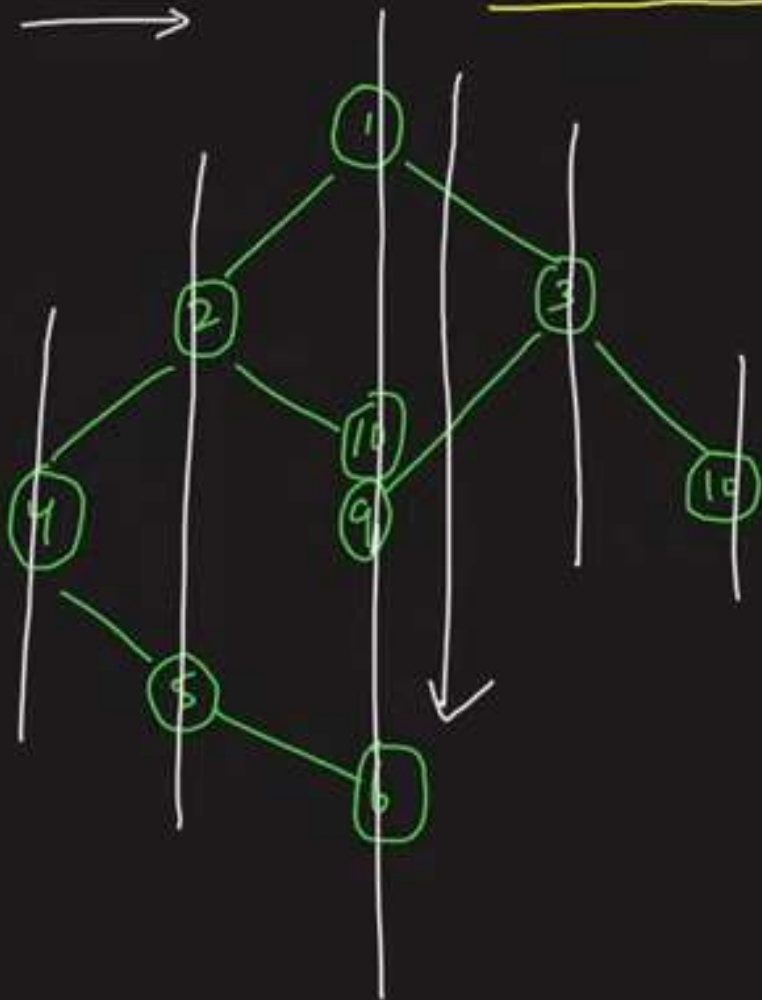


Vertical order traversal



→ 4

→ 2 5

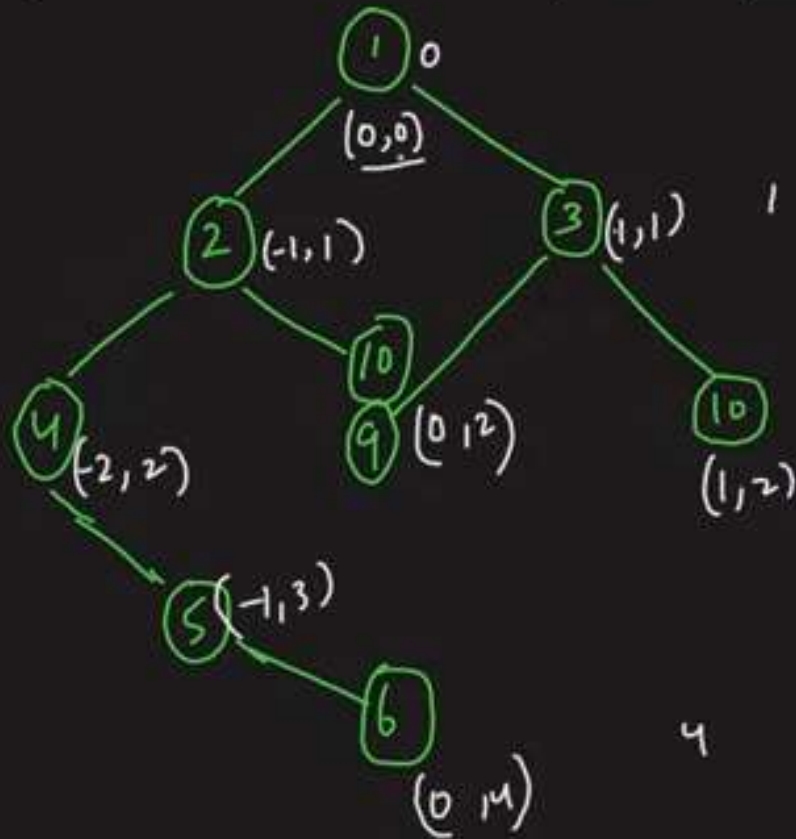
→ [1 9 10 6]

→ 3

→ 10

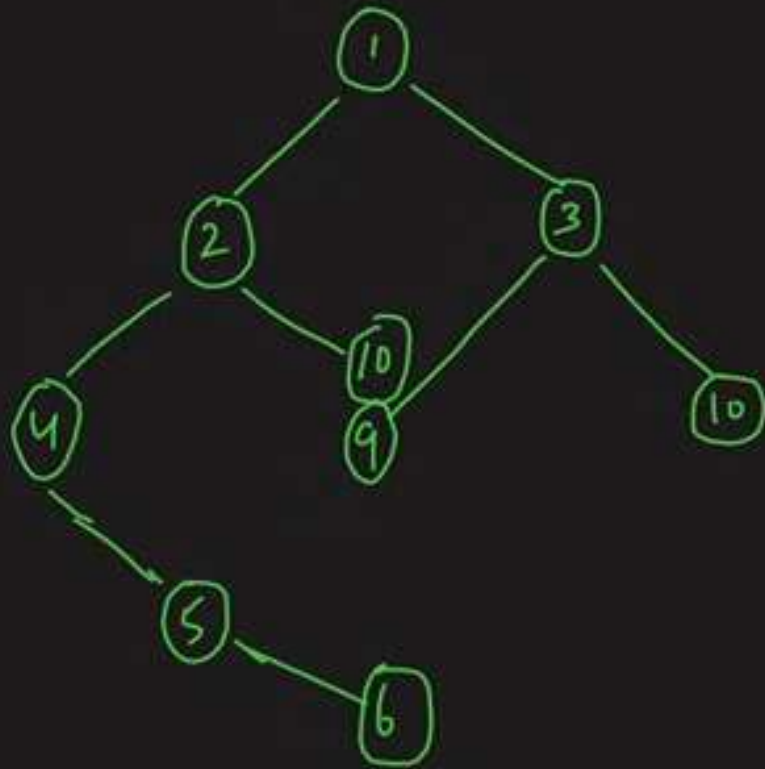
Vertical order traversal

-2 -1 0 +1 +2



2
3
level ↓

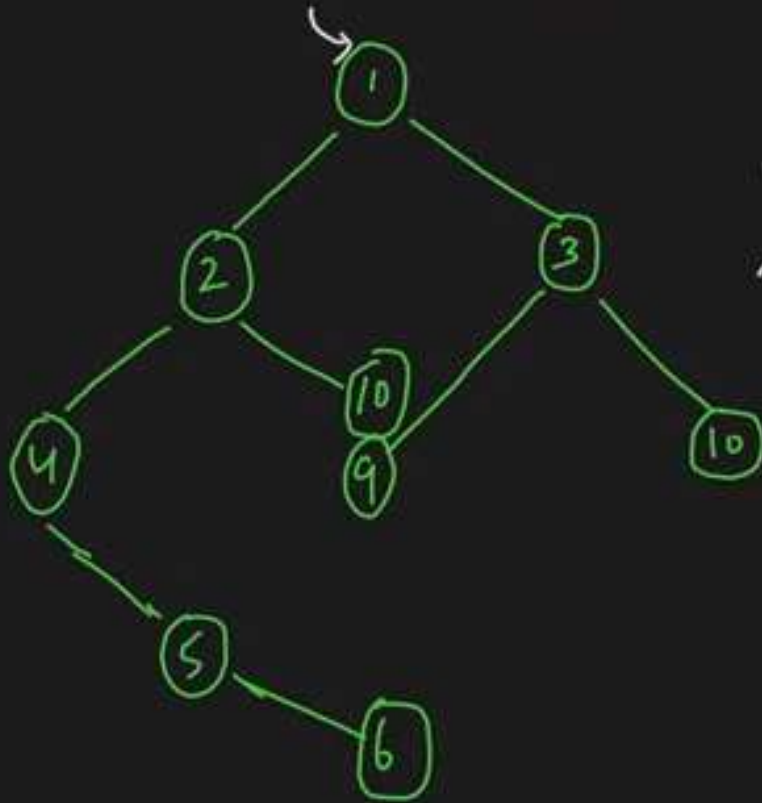
4
2 5
1 9 10 6



$\Theta(\text{node}, v, \text{level})$

$\text{map} \langle \text{nt}, \text{map} \langle \text{nt}, \text{multiset} \langle \text{nt} \rangle \rangle \rangle$
 vertical level multi nodes
 $\text{Treemap} \langle \text{nt}, \text{Treemap} \langle \text{nt}, \text{po} \rangle \rangle$

Vertical order traversal

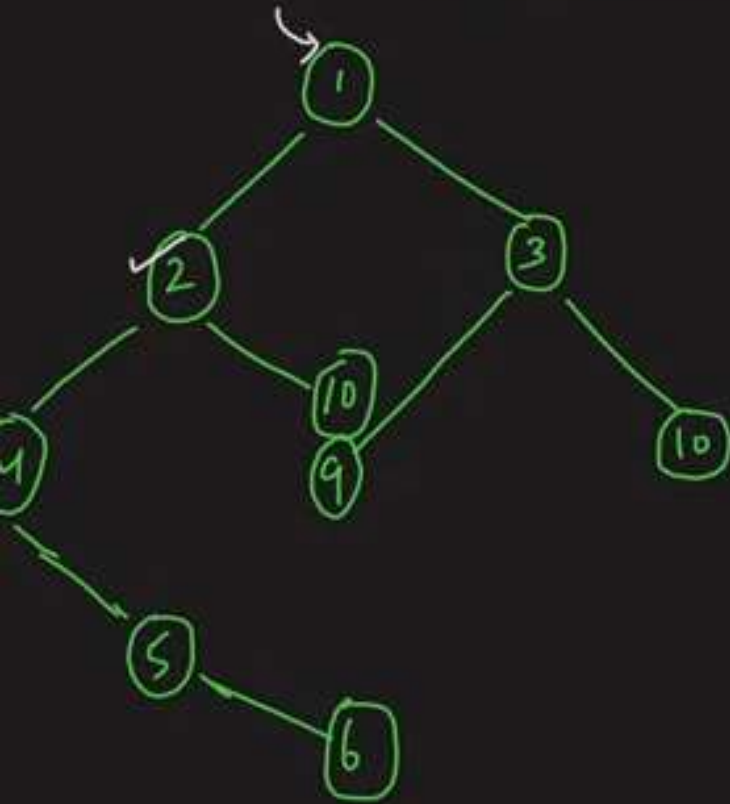


node = 1 (0, 0)

(1, 0, 0) (2, -1, 1) (3, 1, 1)
 $\Theta(\text{node}, v, \text{level})$

0 \rightarrow 0, \rightarrow {1}

Vertical order traversal

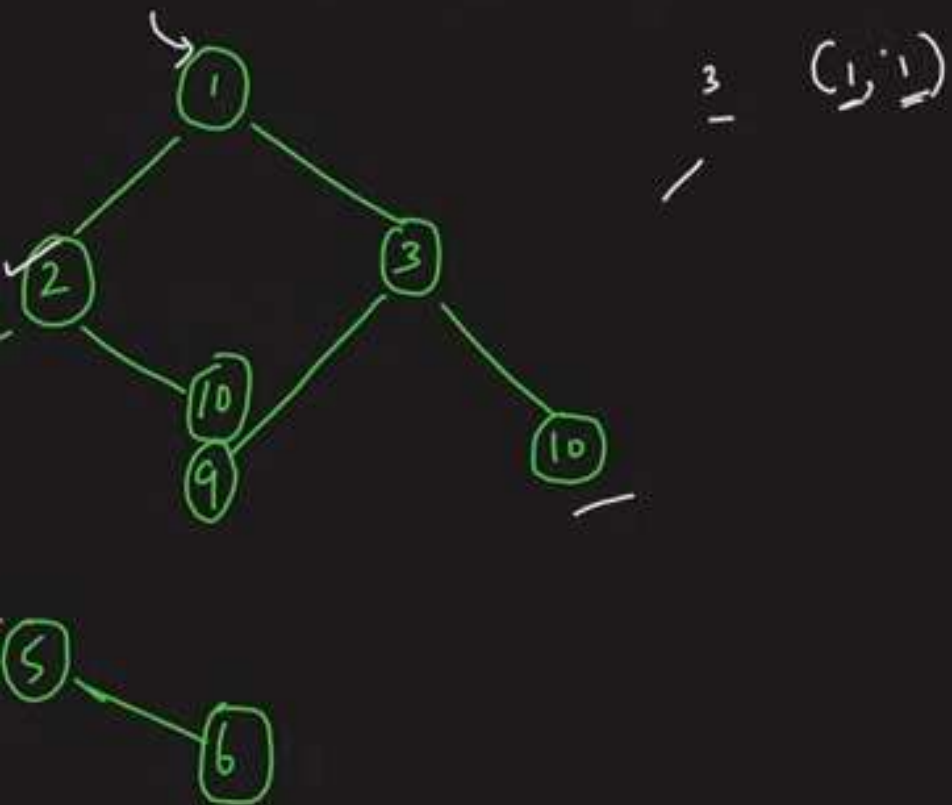


node = 2 $(-1, 1)$

$\boxed{(2, -1, 1) (3, 1, 1) (4, -2, 2) (10, 0, 2)}$
 $\Theta(\text{node}, v, \text{level})$

$-1 \rightarrow 1 \rightarrow \{2\}$
 $0 \rightarrow 0 \rightarrow \{1\}$

Vertical order traversal



$$\boxed{(\cancel{3}, +, +) (4, -2, 2) (10, 0, 2) (9, 0, 2)}$$

$$\Theta(\text{node}, v, \text{level})$$

(10, 2, 2)

$$\begin{aligned} -1 &\rightarrow 1 \rightarrow \{2\} \\ 0 &\rightarrow 0 \rightarrow \{1\} \\ 1 &\rightarrow 1 \rightarrow \{3\} \end{aligned}$$

Get me steps by steps when and what things are getting added

Thought for 16s

Awesome—let's walk through **exactly** what gets added, step by step, using this tree:

markdown

Copy Edit

```

      3
     / \
    9  20
     / \
    15  7
```

Coordinates:

- root 3 at $(x=0, y=0)$
- left = $x-1, y+1$, right = $x+1, y+1$

We maintain:

- `q` = BFS queue of Tuples (node, x, y)
- `map` = `TreeMap<Integer, TreeMap<Integer, PriorityQueue<Integer>>>`
 - keys: `x` \rightarrow column
 - inner keys: `y` \rightarrow row
 - value: `PQ` of node values at same (x,y)



Initialization

- `map = { }`
 - `q = [(3, 0, 0)]`
-

Iteration 1

Dequeue: `(3, 0, 0)`

1. Insert into `map`
 - `x=0` not present \Rightarrow create inner `TreeMap`
 - `y=0` not present under `x=0` \Rightarrow create `PriorityQueue`
 - offer value `3` into `map[0][0]`
2. Enqueue children
 - left child `9` \Rightarrow `(x=-1, y=1)`
 - right child `20` \Rightarrow `(x=1, y=1)`

State after step

- `map = { 0: { 0: [3] } }`
 - `q = [(9, -1, 1), (20, 1, 1)]`
-

Iteration 2

Dequeue: `(9, -1, 1)`



Iteration 2

Dequeue: (9, -1, 1)

1. Insert

- $x=-1$ not present \Rightarrow create inner `TreeMap`
- $y=1$ not present \Rightarrow create `PQ`
- offer 9 into `map[-1][1]`

2. Enqueue children

- 9 has no children \Rightarrow nothing added

State after step

- `map = { -1: { 1: [9] }, 0: { 0: [3] } }`
- `q = [(20, 1, 1)]`

Iteration 3

Dequeue: (20, 1, 1)

1. Insert

- $x=1$ not present \Rightarrow create inner `TreeMap`
- $y=1$ not present \Rightarrow create `PQ`
- offer 20 into `map[1][1]`

2. Enqueue children

- left 15 \Rightarrow `(x=0, y=2)`
- right 7 \Rightarrow `(x=2, y=2)`



State after step

- `map = { -1: { 1: [9] }, 0: { 0: [3] }, 1: { 1: [20] } }`
 - `q = [(15, 0, 2), (7, 2, 2)]`
-

Iteration 4

Dequeue: `(15, 0, 2)`

1. Insert

- `x=0` exists
- `y=2` not present under `x=0` \Rightarrow create PQ
- offer `15` into `map[0][2]`

2. Enqueue children

- none

State after step

- `map = { -1:{1:[9]}, 0:{0:[3], 2:[15]}, 1:{1:[20]} }`
 - `q = [(7, 2, 2)]`
-

Iteration 5

Dequeue: `(7, 2, 2)`

1. Insert

- `x=2` not present \Rightarrow create inner `TreeMap`



1. Insert



- `x=2` not present \Rightarrow create inner `TreeMap`
- `y=2` not present \Rightarrow create `PQ`
- offer `7` into `map[2][2]`

2. Enqueue children

- none

Final state after BFS

yaml

 Copy  Edit

```
map = {  
  -1 : { 1 : [9] },  
  0 : { 0 : [3], 2 : [15] },  
  1 : { 1 : [20] },  
  2 : { 2 : [7] }  
}  
q = []
```



Collection Phase (building the answer)

Read columns (`x`) in sorted order, and within each, rows (`y`) in sorted order; pop each PQ:

- `x = -1` \rightarrow rows: `1` \rightarrow `[9]`
- `x = 0` \rightarrow rows: `0, 2` \rightarrow `[3, 15]`
- `x = 1` \rightarrow rows: `1` \rightarrow `[20]`



```
1 : { 1 : [20] },  
2 : { 2 : [7] }  
}  
q = []
```

 Copy  Edit

Collection Phase (building the answer)

Read columns (`x`) in sorted order, and within each, rows (`y`) in sorted order; pop each PQ:

- `x = -1` → rows: `1` → `[9]`
- `x = 0` → rows: `0, 2` → `[3, 15]`
- `x = 1` → rows: `1` → `[20]`
- `x = 2` → rows: `2` → `[7]`

Result: `[[9], [3, 15], [20], [7]]`