Properties: for every node n, values in left subtree of n < value in n right > value in n
**Interval Tree**

\[ \text{isset\_contains}(s, 14) \]

- root: \[ 6-10 \]
- size: \[ 3 \]

```
found
n
parent
dir
last\_left
last\_right
```

```
2-2
```

```
19-19
```

```
14-15
```

```
23-25
```

```
9
```

```
1m \rightarrow 2m
1
2
4 \rightarrow 100-100000
```

Inorder Traversal:

- next\_excluded \((s, 9) = 11\)
- count\_interval \((s) = 5\)
- next\_excluded \((s, 20) = 20\)
- size \((s) = 13\)
isset-add(5, 21)

1) make new node
2) set parent from new node
3) set child from parent
4) update bookkeeping

isset-add(5, 20)
isset_add(5, 21)

1) make new node
2) set parent from new node
3) set child from parent
4) update bookkeeping

isset_add(5, 20)
merge with last_left or last_right
AVL Tree

Balance: for each node,
\[ \text{balance} = \text{height of left} - \text{height of right} \]

AVL Tree: balance all \(-1, 0, 1\)

\[ \text{height} \approx O(\log n) \]

\text{add}(s, yvr)

Oct 30 Page 5