Query: given an interval $\times$, return any interval in the set $S$ that partially overlaps $\times$ (if one exists)
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: \( l_0[s_i] \) vs \( L \)
- is there some large enough \( h_i[s_i] \)?
- is there some small enough \( l_0[s_i] \)?

If \( l_0[s_i] \leq L \) and \( h_i[s_i] > L \), then overlap
If \( l_0[s_i] > L \) and \( L \leq R \), then overlap
SEARCHING FOR OVERLAPPING INTERVALS

Augment: MAX RIGHT END of SUBTREE

BST w/ LEFT ENDS as KEYS

ID:
SEARCHING FOR OVERLAPPING INTERVALS

ID:

compare w/ root first

query segment
SEARCHING FOR OVERLAPPING INTERVALS

1D:

\[ R < x < w \]

Case 1: IF NO OVERLAP

right subtree can't overlap

keep searching
LEFT
SEARCHING FOR OVERLAPPING INTERVALS

**1D:**

**IF \( Z \geq L \)**

- search left

**IF NO OVERLAP**

- case 2

**\( y \leq z' \)**

- guaranteed overlap
SEARCHING FOR OVERLAPPING INTERVALS

IF $Z \geq L$
- search left

IF NO OVERLAP
- case 2

$\exists y^2$
- guaranteed overlap

s.t. $y < L < z'$

else ($Z < L$)
- no overlap to left
- search right
How can we update MAX RIGHT END of SUBTREE?

BST w/ LEFT ENDS as KEYS
augmented BST

$$\text{max}(t) = \max \begin{cases} h_i(t) \\ \text{max}(t_L) \\ \text{max}(t_R) \end{cases}$$
max1, max2, max3: unchanged by rotation
max(A) & max(B): trivial to update

we can maintain a balanced BST augmented w/ max value of subtrees