set $S$ of intervals

$4 \quad \quad \quad \quad \quad \quad 8$

$5 \quad \quad \quad \quad \quad \quad 11$

$7 \quad \quad \quad \quad \quad \quad 10$

$15 \quad \quad \quad \quad \quad \quad 18$

$17 \quad \quad \quad \quad \quad \quad 19$

$21 \quad \quad \quad \quad \quad \quad 23$
Query: given an interval $\times$, return any interval in the set $S$ that partially overlaps $\times$ (if one exists)
types of overlap:
i) "smaller"
types of overlap:
1) "smaller"
2) "bigger"
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: $l_0[s_i]$ vs $L$
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: $lo[s_i]$ vs $L$

is there some large enough $hi[s_i]$?
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: $l_0[s_i]$ vs $L$

is there some large enough $hi[s_i]$?

If $l_0[s_i] \leq L$ AND $hi[s_i] \geq L$ then overlap
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: $l_0[s_i]$ vs $L$

is there some small enough $l_0[s_i]$?

If $l_0[s_i] \geq L$ AND $\leq R$ then overlap
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

First comparison: $lo[s_i]$ vs $L$
- is there some large enough $hi[s_i]$?
- is there some small enough $lo[s_i]$?

If $lo[s_i] \leq L$ and $hi[s_i] > L$, then overlap.
If $lo[s_i] > L$ and $\leq R$, then overlap.
Store a balanced BST on lo[si]
Store a balanced BST on lo[si]

Compare L to nodes,
splits into 2 groups: < & >
Store a balanced BST on lo[si]

Compare L to nodes,
splits into 2 groups: < & >
SEARCHING FOR OVERLAPPING INTERVALS

ID:

BST w/ LEFT ENDS as KEYS
SEARCHING FOR OVERLAPPING INTERVALS

ID:

BST w/ LEFT ENDS as KEYS

MAX RIGHT END OF SUBTREE
SEARCHING FOR OVERLAPPING INTERVALS

ID:

compare w/ root first

query segment
SEARCHING FOR OVERLAPPING INTERVALS

IF NO OVERLAP

CASE 1

L < x

R < x
SEARCHING FOR OVERLAPPING INTERVALS

IF NO OVERLAP

right subtree can't overlap

case 1

R < x < W
SEARCHING FOR OVERLAPPING INTERVALS

ID:

---

CASE 1

IF NO OVERLAP

right subtree can't overlap

keep searching

LEFT

R < x < W
SEARCHING FOR OVERLAPPING INTERVALS

IF NO OVERLAP

L case 2 R
SEARCHING FOR OVERLAPPING INTERVALS

IF \( Z \gg L \)

IF NO OVERLAP

\[ \text{case 2} \]

L \quad R
SEARCHING FOR OVERLAPPING INTERVALS

1D:

IF $Z \geq L$

- search left

IF NO OVERLAP

$\exists y' z'$

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

$\{\text{guaranteed overlap}\}$

case 2

$L \quad R$
SEARCHING FOR OVERLAPPING INTERVALS

1D:

IF $Z \geq L$
- search left

IF NO OVERLAP
- case $L \leq z \leq R$

$\exists y \in (y, y')$
- s.t. $y < L < y'$
- guaranteed overlap

else $(z < L)$
SEARCHING FOR OVERLAPPING INTERVALS

IF \( Z \geq L \) search left

IF NO OVERLAP

Case 2

\( L \) \quad \text{R}

\exists y, z' \quad \begin{cases} \text{guaranteed overlap} \\ \text{else (} z < L \text{)} \end{cases}

\cdot \text{no overlap to left}
\cdot \text{search right}
How can we update the maximum right end of a subtree?

BST w/ left ends as keys
BST on lo[s;]
augmented BST

max(t) = max \left\{ h_i(t), \max(t_L), \max(t_R) \right\}
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