**Interval Trees**

Set $S$ of intervals:

- $(4, 8)$
- $(5, 11)$
- $(15, 18)$
- $(17, 19)$

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

First comparison: $lo[s_i]$ vs $lo[x]$
types of overlap:
1) “smaller”
2) “bigger”
3) “left” & “right”

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

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

If $lo[s_i] < lo[x]$
AND $hi[s_i] > lo[x]$
then overlap
types of overlap:
1) "smaller"
2) "bigger"
3) "left" & "right"

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

If $lo[s_i] < lo[x]$  
AND  
$hi[s_i] > lo[x]$  
then overlap

If $lo[s_i] > lo[x]$  
AND  
$\leq hi[x]$  
then overlap
First comparison: $lo[s_i] \text{ vs } lo[x]$

- is there some large enough $hi[s_i]$?
- is there some small enough $lo[s_i]$?

If $lo[s_i] \leq lo[x] \text{ AND } hi[s_i] > lo[x]$ then overlap

If $lo[s_i] > lo[x] \text{ AND } \leq hi[x]$ then overlap
Store a balanced BST on $lo[s_i]$

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

Compare $lo[x]$ to nodes, splits into 2 groups: $<$ & $>$

if $lo[x] < lo[s_j]$ (and $hi[x] < lo[s_j]$)

Notice that $lo[s_j] \leq lo[si]$ for the entire subtree of nodes to the right ...
...go left.

if $lo[x] > lo[sk]$

(and $lo[x] > hi[sk]$)

compare $lo[x]$ to max of $hi[si]$ of the entire subtree of nodes to the left stored at $sk$ (augment)

depends
SEARCHING FOR OVERLAPPING INTERVALS

BST w/ LEFT ENDS as KEYS

MAX RIGHT END OF SUBTREE
SEARCHING FOR OVERLAPPING INTERVALS

ID:

IF NO OVERLAP
right subtree can't overlap

keep searching
LEFT

R < x < W
SEARCHING FOR OVERLAPPING INTERVALS

1D:

IF \( z \geq L \)

search left

\( \exists y' \)

s.t.

\( y < l < z' \)

\{ guaranteed overlap \}

IF NO OVERLAP

\( \text{case 2} \)

L

R
SEARCHING FOR OVERLAPPING INTERVALS

IF $Z \geq L$
- Search left

$\exists yz' \ s.t. \ y < l < z'$
- Guaranteed overlap

IF NO OVERLAP

L case 2 R

else $(Z < L)$
- No overlap to left
- Search right
augmented BST

\[ \max(t) = \max \left\{ h_i(t), \max(t_L), \max(t_R) \right\} \]
$\text{max1, max2, max3: unchanged by rotation}$

$\text{max(A) & max(B): trivial to update}$

$\text{we can maintain a balanced BST augmented w/ max value of subtrees}$
Find-Overlap\( (x,S) \)

\[ y \leftarrow \text{root interval in } S \]

while \( y \neq \emptyset \) & \( \begin{cases} \text{no overlap} & \text{or} \\ \text{lo}(x) > \text{hi}(y) & \text{or} \\ \text{lo}(y) > \text{hi}(x) & \end{cases} \]

\[ \{ \text{while no trivial solutions:} \ (\text{no overlap}) \ - \text{OR-} \ (xy \text{ overlap}) \]  

return \( y \)
Find-Overlap\((x, S)\)

\[
y \leftarrow \text{root interval in } S
\]

while \(y \neq \emptyset\)  

\[
\left\{ \begin{array}{l}
\text{if } \lo(x) > \hi(y) \\
\text{or} \\
\lo(y) > \hi(x)
\end{array} \right.
\]

\} while no trivial solutions: (no overlap) - OR - (xy overlap)

\[
\text{if } y_L \neq \emptyset \text{ and } \lo(x) \leq \max(y_L)
\]

\[
\text{then } y \leftarrow y_L
\]

else \(y \leftarrow y_R\)  

\]

\[
\text{return } y
\]

For node \(p:\)

- left/right child = \(P_L, P_R\)

- if no child, use \(\emptyset\)

- \(\max(p) = \max(\hi[\text{subtree}(p)])\) in subtree\((p)\)
Time = height = $O(\log n)$

All $k$ overlaps: ?