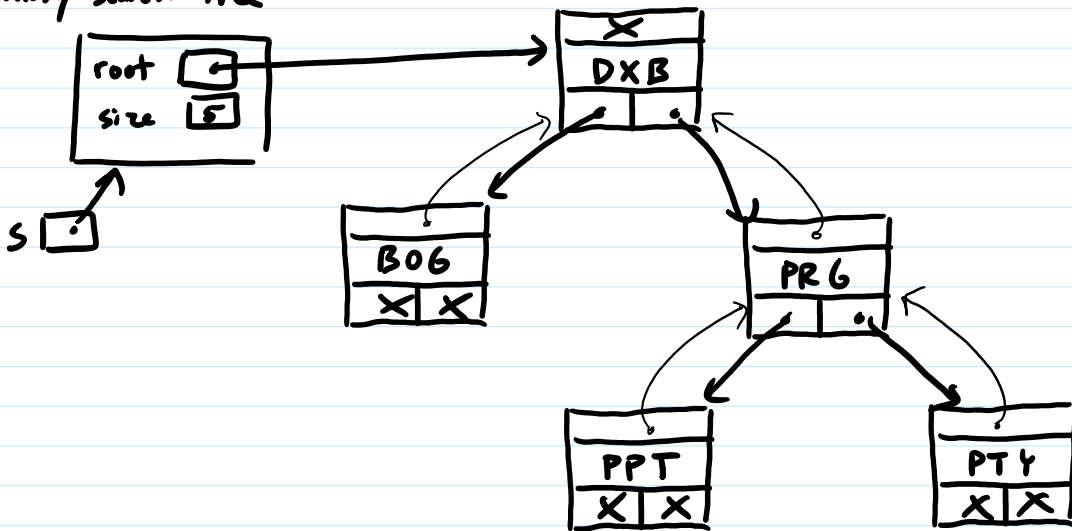


Binary Search Tree

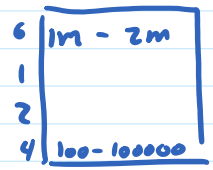
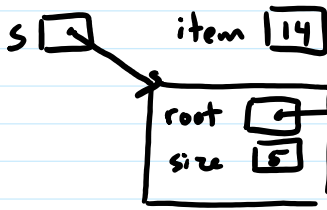


Properties: for every node n , values in left subtree of $n <$ value in n
right $>$ value in n

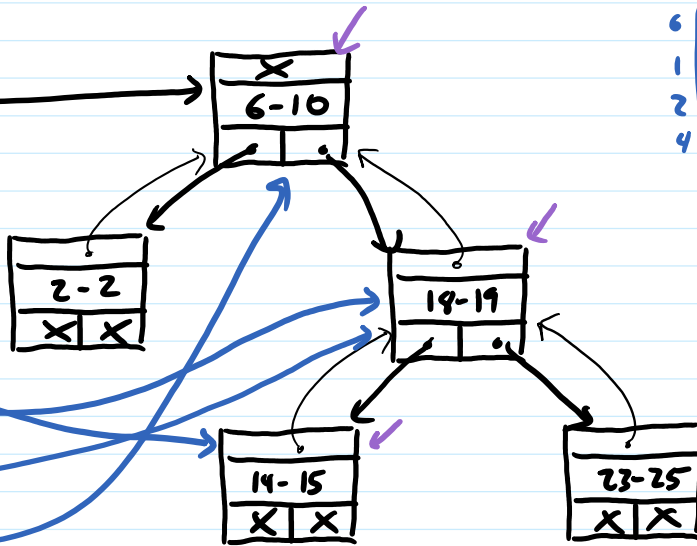
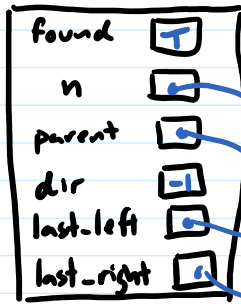
Interval Tree

100 - 100000 1000000 - 2000600

isset-contains(s, 14)



find 500?
have to search
entire hash
table



next_excluded(s, 4) = 11

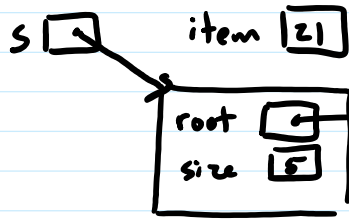
count_intervals(s) = 5

Inorder Traversal:

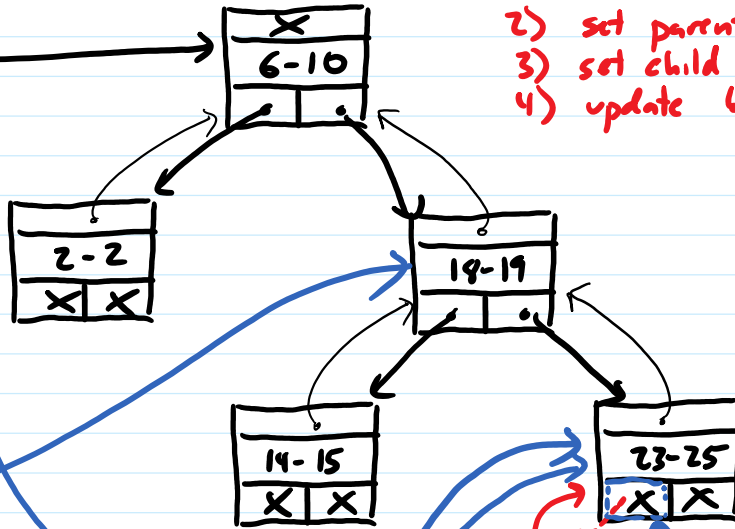
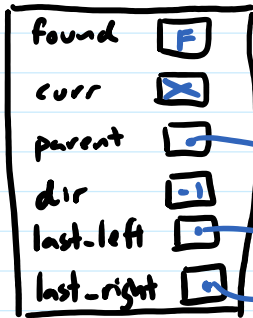
next_excluded(s, 20) = 20

size(s) = 13

isset-add (s, 21)



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

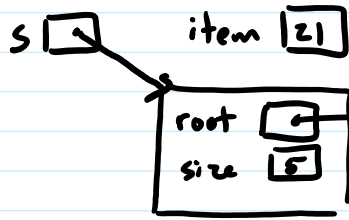


incoming

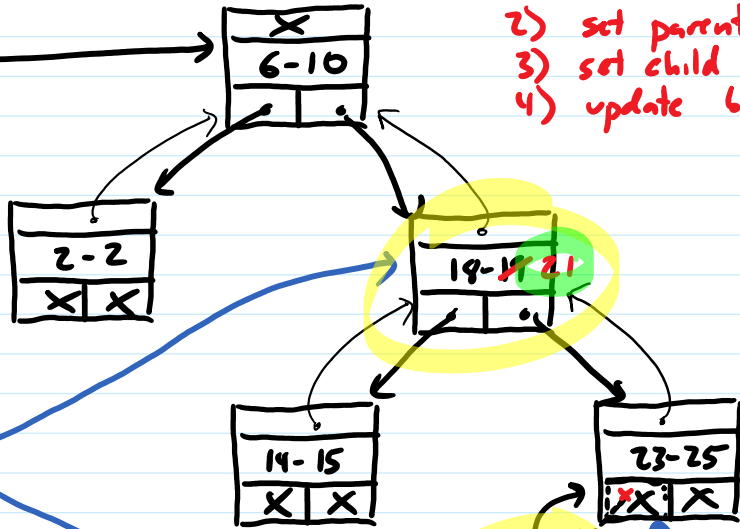
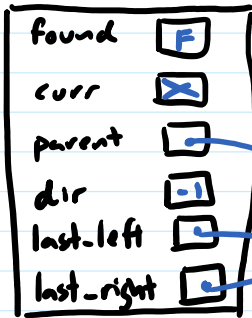
* incoming = new-node

isset-add(s, 20)

isset-add (s, 21)



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



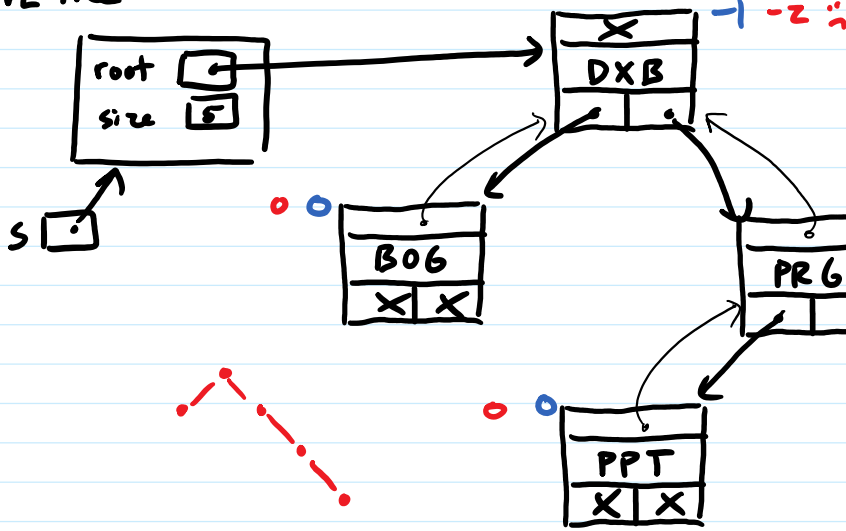
incoming

* incoming = new-node

merge with last-left or last-right

isset-add(s, 20)

AVL Tree



balance: for each node,
height of left
-
height of right

AVL tree:
balances all $-1, 0, 1$
height $O(\log n)$

add(s, YVR)

