Review binary search trees

search j
search g

insertion is a search that ends at "null" + place it.
Finding predecessor

Note \( q \leq V \leq X \leq S \)

Find pred. of \( X \)

Case 1: \( T \) exists

\( V \) is pred

Case 2: \( T \) empty

Then \( q \) is pred.

only \( V \) could be pred (rest are smaller)
Case 3.

T empty \& q doesn't exist

No predecessor

x is first

in an in order traversal

Successor is symmetric
Delete

easy case - delete a leaf
  Remove it

medium case - delete a node with one child

null could be replaced by right child

splice it out

\[ \Rightarrow \]
Hardest case delete a node with 2 children could use succ pred has 0 or 1 children remove pred from left subtree