Lecture 11: Introduction to Binary Search Trees
Warm-Up

Insert into an initially empty BST: 50, 25, 75, 60, 55, 90, 65, 37
Delete 62
Delete 17
Deletion from a Binary Search Tree

Delete 78
Print Keys in order

Write a function:
- Print each key
- Print smallest to largest

\[
\text{in-order}(\text{Node } r) \\
\text{if } (\text{nullptr} == r) \\
\text{return} \\
\text{in-order}(r->\text{left}) \\
\text{out } r->\text{key} \\
\text{in-order}(r->\text{right})
\]
Rerecreate the Tree

Write a function:

- Print each key
- Allow me to recreate the tree

```
pre-order (Node * r)
if (r == nullptr) return;
out r->key
pre-order (r->left);
pre-order (r->right);
```
Syntax Trees

Binary, not search \( ((x + y)^2) + ((x - 4)/3) \).
Draw this tree

- Every non-null node is one character