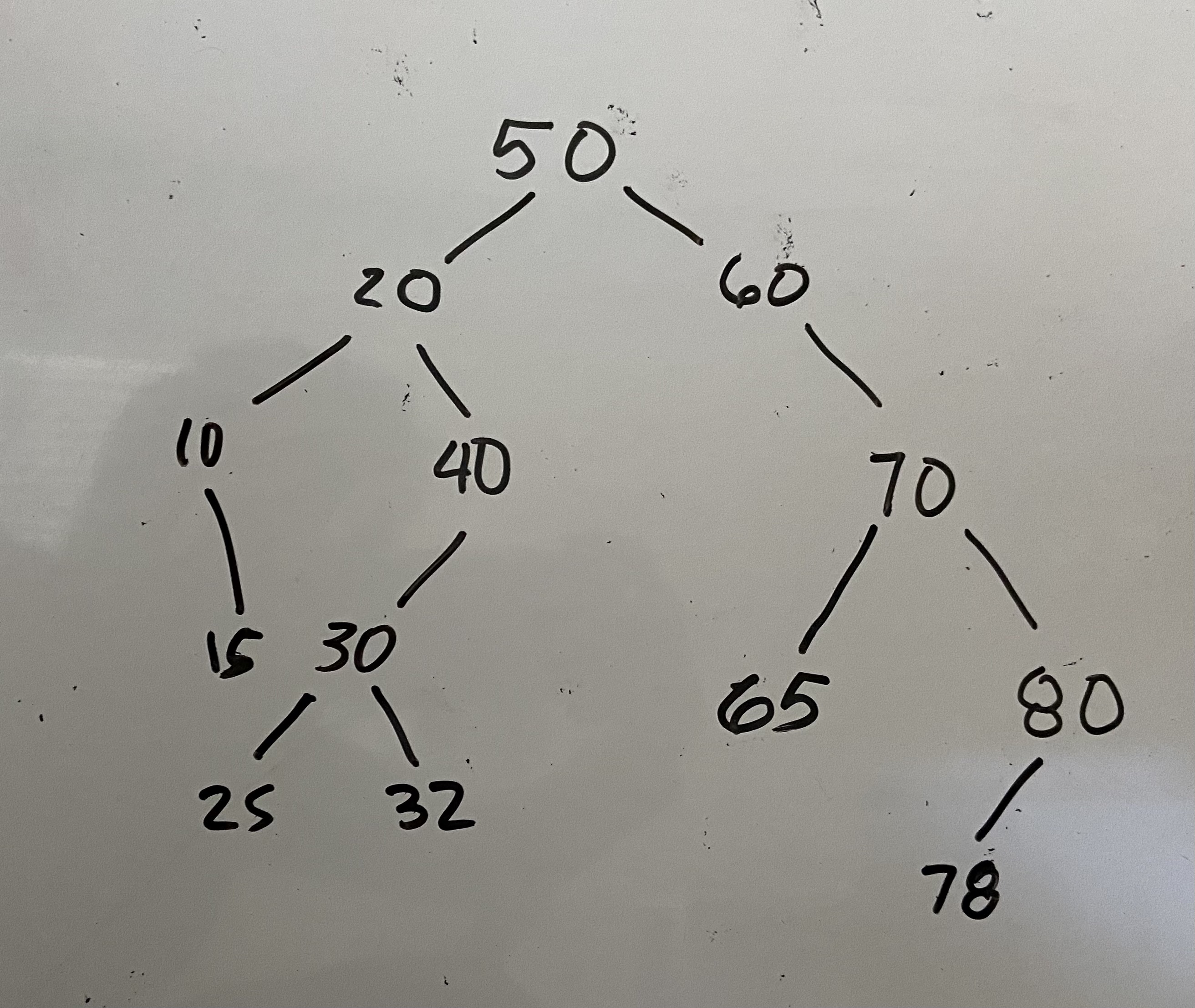
**Chelsea Biala**

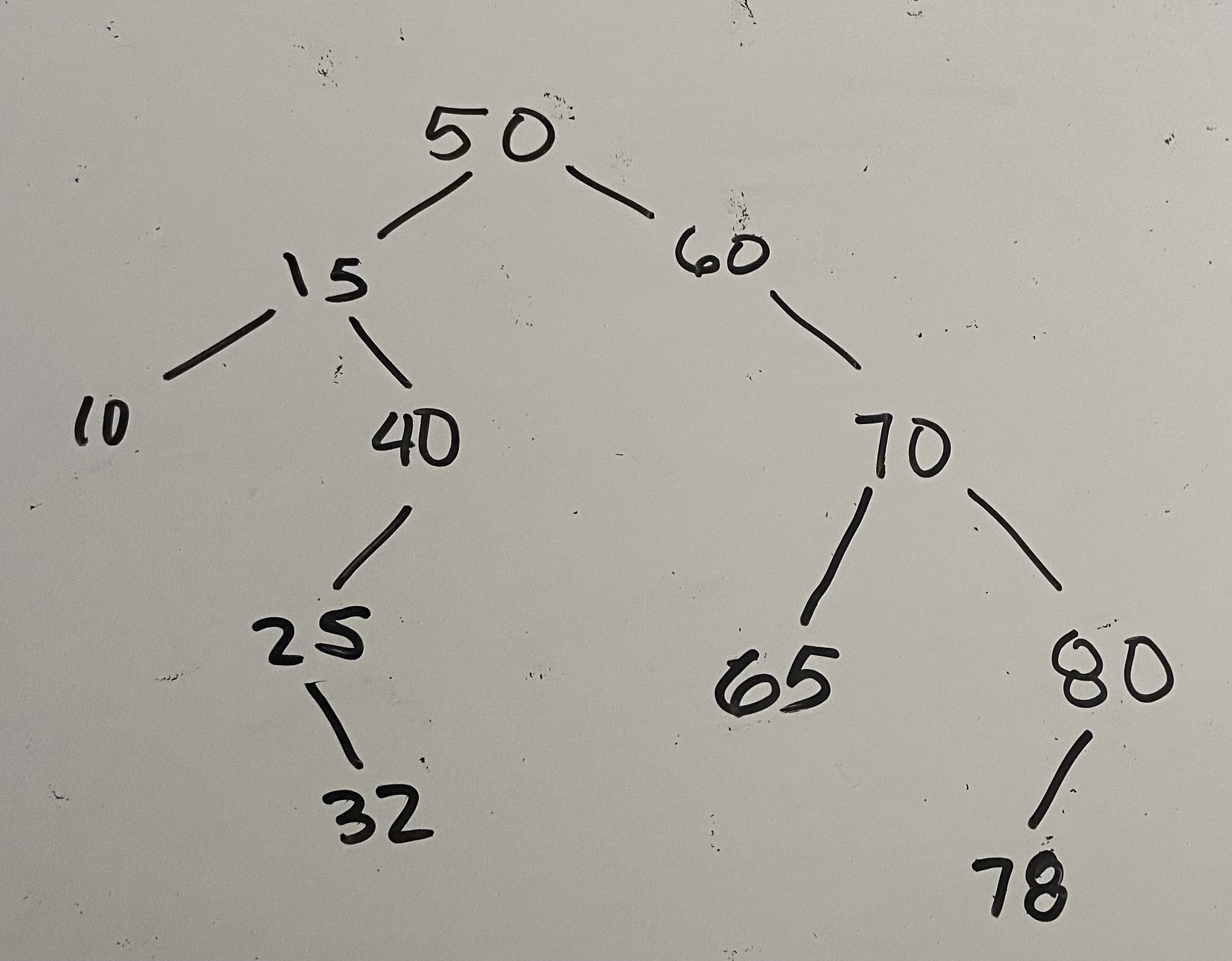
**CS32 Lecture 2 Smallberg**

1a.

1b. InOrder: 10 15 20 25 30 32 40 50 60 65 70 78 80

PreOrder: 50 20 10 15 40 30 25 32 60 70 65 80 78

PostOrder: 15 10 25 32 30 40 20 65 78 80 70 60 50

1c.

2a. struct Node

{

private:

int data;

Node\* leftChild;

Node\* rightChild;

Node\* parent;

}

2b. insert(int d)

if tree is empty

create new node with data d

root pointer points to new node

start at root of tree

while not done

if d is equal to current node’s value

done

if d is less than current node’s value

if there is a left child

set parent to current and move current to left child

else

create new node with data d, set current’s left pointer to new node, set new node’s parent to current

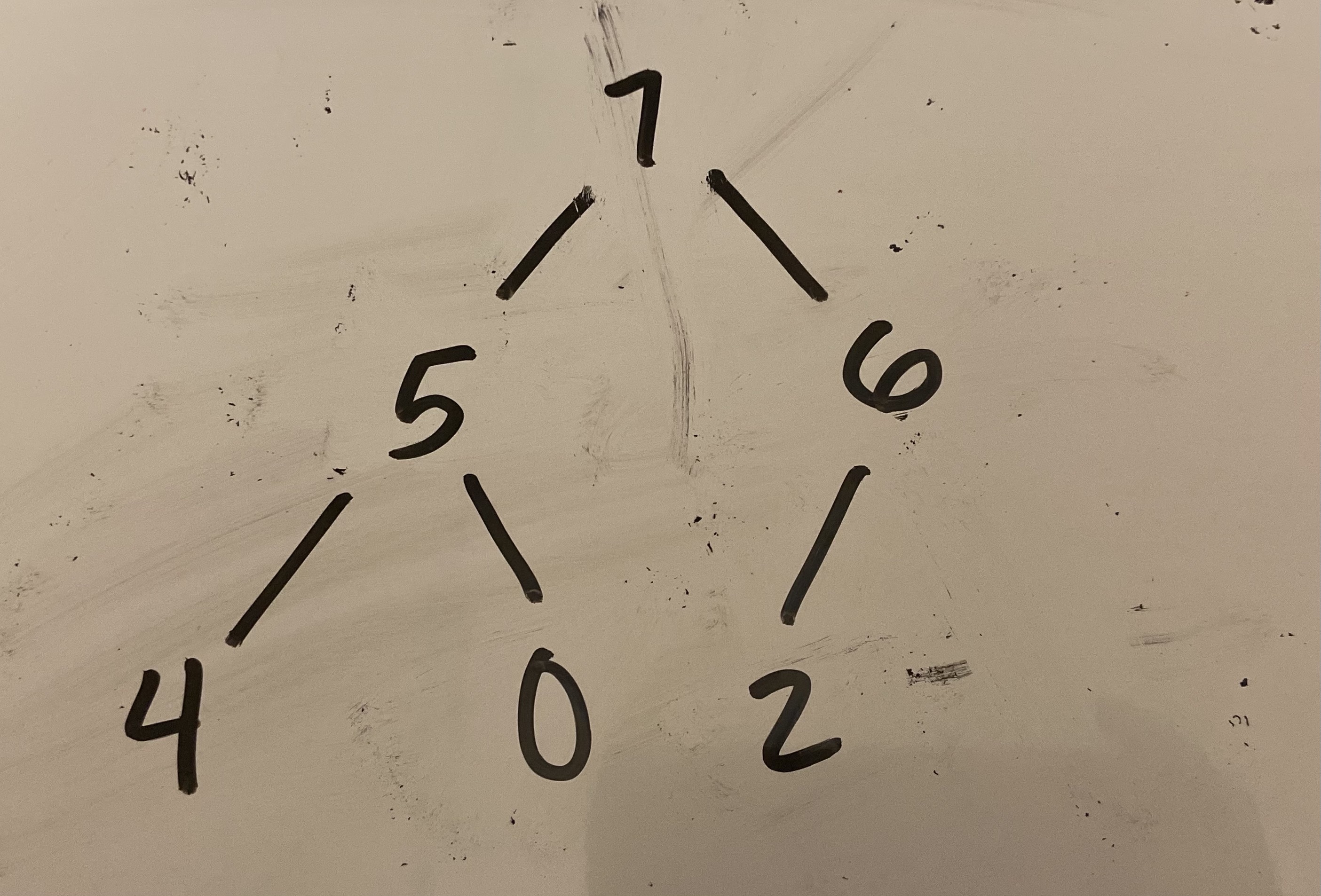
if d is greater than current node’s value

if there is a right child

set parent to current and move current to right child

else

create new node with data d, set current’s right pointer to new node, set new node’s parent to current

3a.

3b. 7 5 6 4 0 2 in that order in positions [0] through [5] of an array

3c. 6 5 2 4 0 in that order in positions [0] through [4] of an array

4a. O(C + S)

4b. O(logC + S)

4c. O(logC + logS)

4d. O(logS)

4e. O(1)

4f. O(logC + S)

4g. O(S)

4h. O(C \* logS)