Case 1: null tree (insert ACGT)

null

becomes

ACGT

Case 2: existing node (insert CAGT)

ACGT

Becomes

Null

ACGT CAGT E E E

Case 3: same start (insert AC)

ACGT

Becomes

Null

Null E E E E

E Null E E E

E E ACGT E AC

Case 4: same sequence (insert ACGT)

ACGT

Becomes

ACGT

Case 5: same sequence different level (insert ACGT)

Null

Null E E E E

E Null E E E

E E ACGT E AC

Becomes

Null

Null E E E E

E Null E E E

E E ACGT E AC

What I think I know:

* After root node, each depth of node-1 = relevant character (depth of 1 looks at first character) (human reading)
* Depth=char for child

**Attempted Algo:**

Returns depth of insertion, otherwise returns -1

Integer insert(String sequence, int depth)- depth starts at 0

If (sequence.equals(root.sequence)- if it equals, return -1

If (root.isLeaf() and root.sequence==null)- set null sequence to sequence and return //base case

If (root.child(charAt(depth)==null)- make a new child if null, find a way to check for $

Make new Node at child of charAt(depth);

Insert(sequence,depth+1);

If (root.isLeaf())

Move(root);

Move(Node root)- moves existing sequence to new place (can be done in insert, separated for now for easier visualization)

String sequence=root.getSequence()

Root.setSequence(null);

Insert(sequence,0);

Assumptions about nodes:

New sequences are null, and new Nodes are leaves

Algo Case 1:

Works

Algo Case 2:

Works

Algo Case 3:

Null

Null E E E E

E Null E E E

E E ACGT E AC