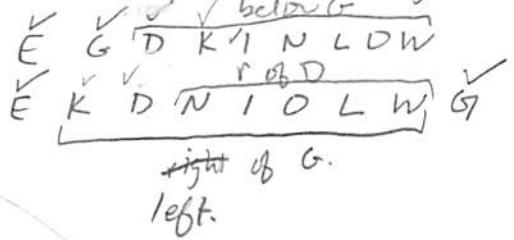
Pre-order: ~~node~~ choose a tree. print the node. move. Every move is an entire left subtree first, then right. print - left - right.

in-order: ~~tree~~ choose a tree. if its left subtree is traversed, print. left - print - right.

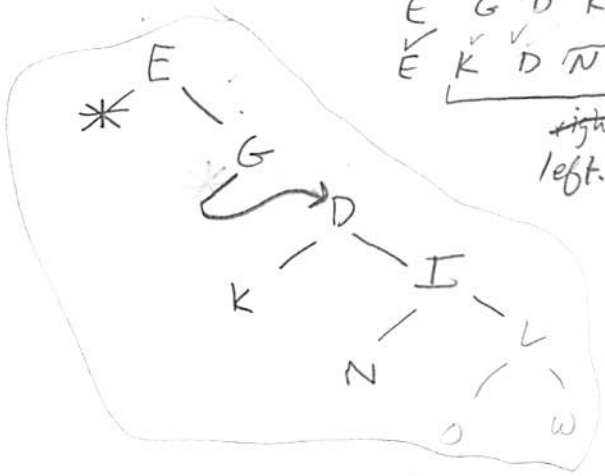
pre-order: when a node is printed, its tree has not been traversed. So all subsequent letters are ~~in the~~ in the node's tree or in a different tree.

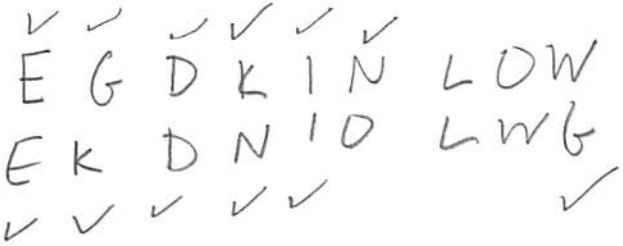
in-order: when a node is printed, its left tree has been traversed. So all subsequent nodes are right or in a diff tree.

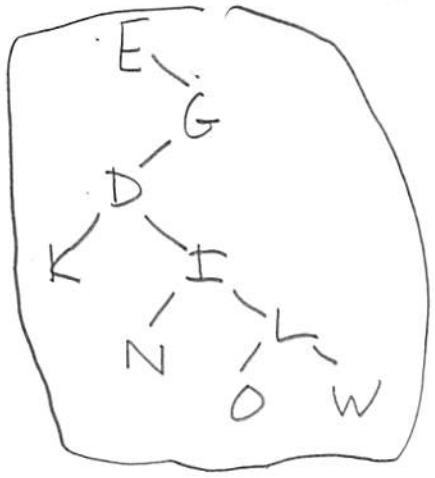


post: left right print

K N O W L I D G E







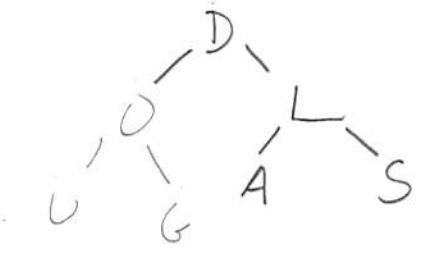
Tree 1

pre – L of D R of D

in-order: U O G D A L S

post U G O A S L D

pre: DOUGLAS



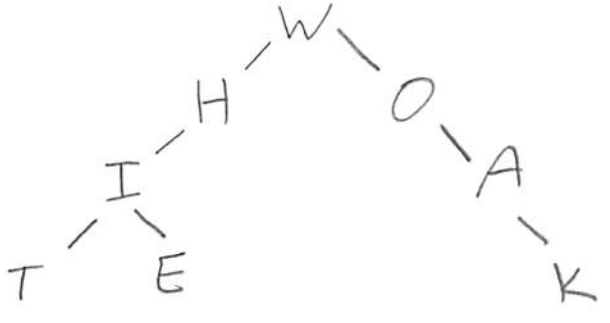
Tree 2

pre: W H I T E O A K print / L / R

in: T E I H K A O W oops

post: T E I H K A O W L / R / print.

in: T I E H W O A K



pre: print / L / R K Č O L M E H

in: L / print / R K Č O L M E H

post: L / R / print. H E M L O C K