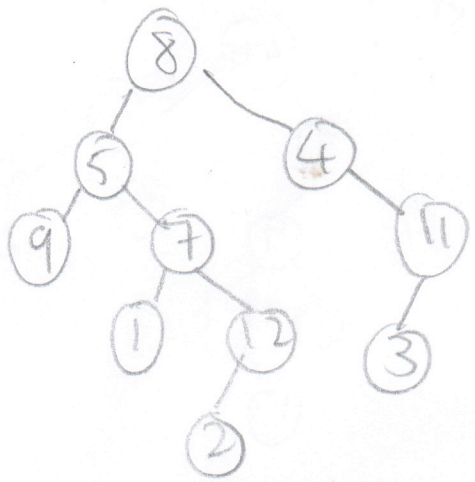
Consider the Following tree



perform the following operations

traversals

pre-order : me 1st, children last
In-order : left → mid → right
post-order : children first, me last

Pre: 8,5,9,7,1,12,2,4,11,3

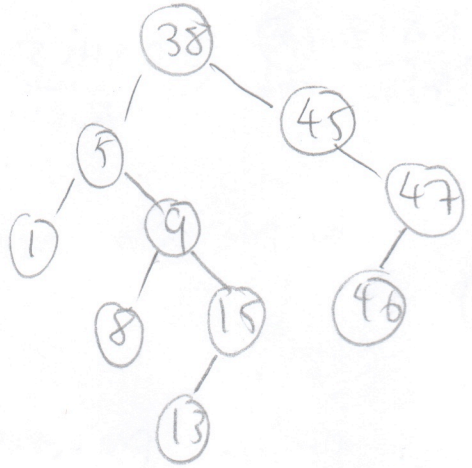
In: 9,5,1,7,2,12,8,4,3,11

post: 9,1,2,12,7,5,3,11,4,8

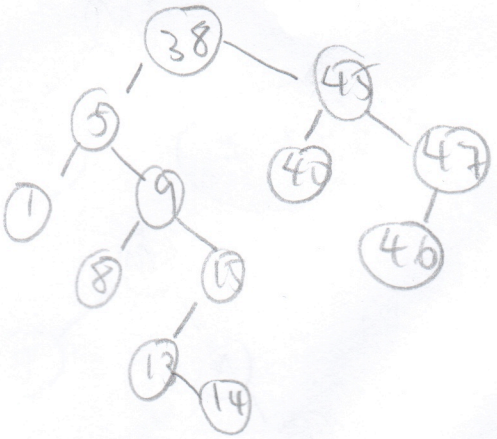
IS this a binary search tree?

NO!

Now given the following tree, perform the following operations

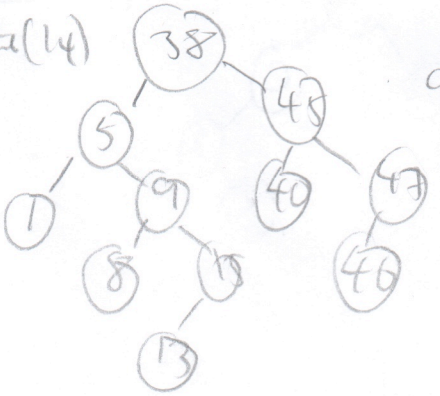


1) Insert (40)
2) Insert (14)

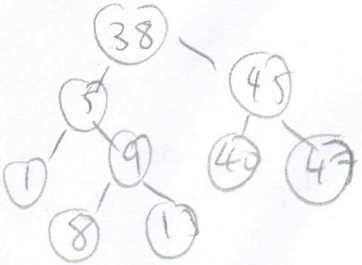


3) delete (14)
4) delete (15)
5) insert (43)
6) delete (38)

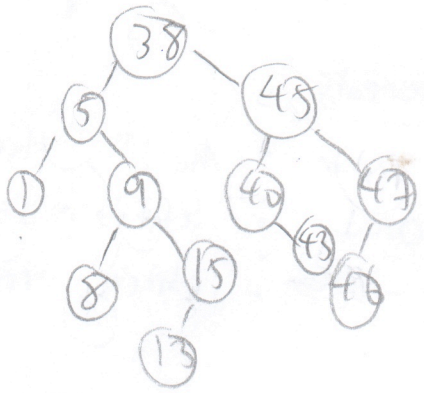
delete (14) →



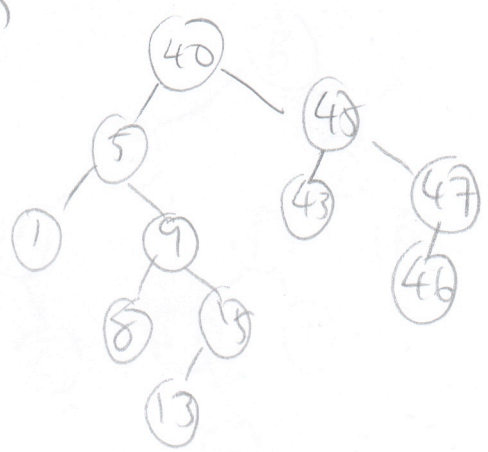
delete (15) →



insert(43)



delete(38) →



Post-order: R I Z A O R S K C H E M

In-order : R A I Z M E O R H C S K

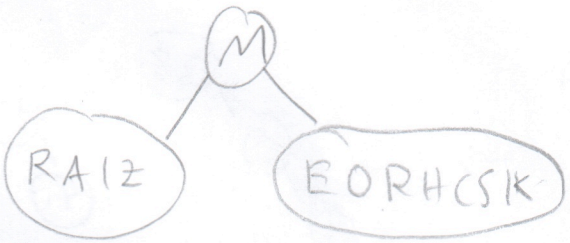
⇒ ∵ post-order : children first, parent last, 'M' is the last visited node

⇒ M must be root of tree

root = (M)

∵ In-order : left → middle → right ⇒ RAIZ (M) EORHCSK

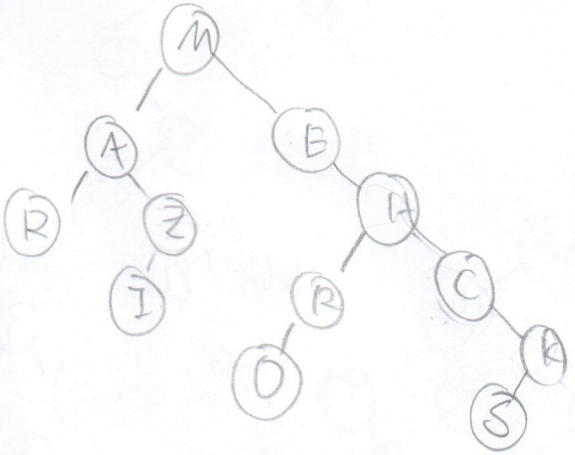
RAIZ: left. M: root. EORHCSK: R.



by some similar reasoning

⋮

⇒



perform pre-order

⇒ MARZIEHROCKS