Assignment 4

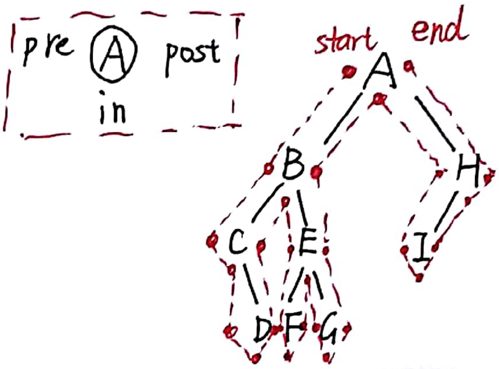
Q1,

(a) A B C D E F G H I

(b) D C FG E B I H A

(c) ~~D~~ C D B F E G A I H

(d) A B H C E I D F G



Along the red line.
The first connected point should show first.

Q2,

Just as Q1 shows, we put three point at left, right and down of one node.
To design a nonrecursive algrithm, I use 3 stacks:

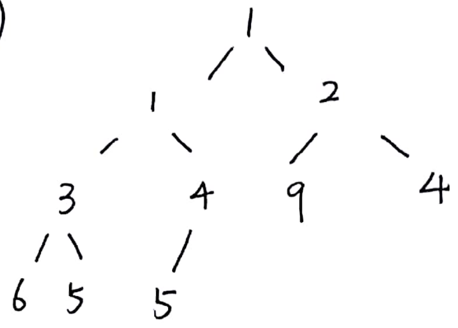
pre-stack,
in-stack,
post-stack,

The program always ① start at the root, ② always asks for left-tree first, ③ if left-tree doesn't exist, then ask for right-tree. ④ if left and right trees both don't exist, back to it's parent.

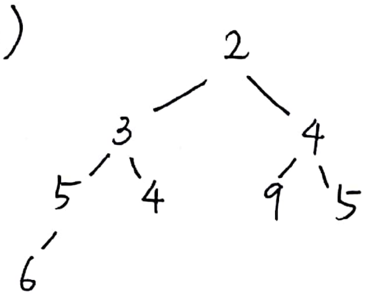
1. we visit one node before ③, then put it into in-stack.
2. we first meet a node, which means it's its parent's left or right tree, we put the node into pre-stack.
3. If a node is in in-stack, then we delete it from ~~in~~ Pre- stack, and add it to post-stack.
4. If a node is leaf, we put it into post-stack.

This algrithm is not only works on In-order traversal, as you can see from the figure at Q1, it works for 3 different traversal, with just a little change.

Q3. (a)



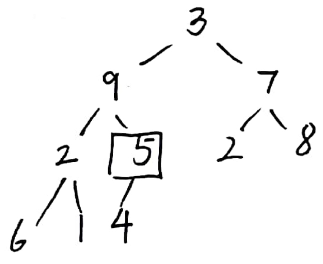
(b)



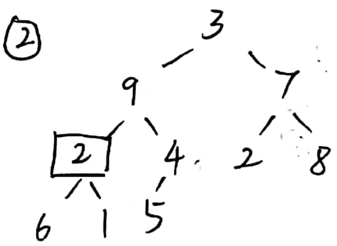
Q4.

① Initializing a Min Heap:

Node at index ⌊10/2⌋ = 5

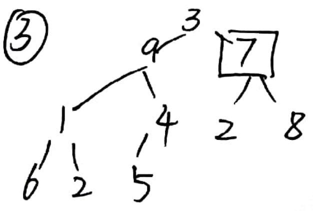


②



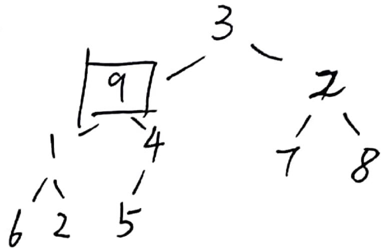
Node at index 4

③



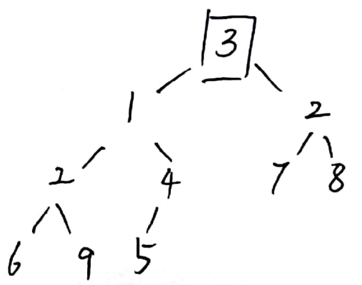
Node at index 3

④



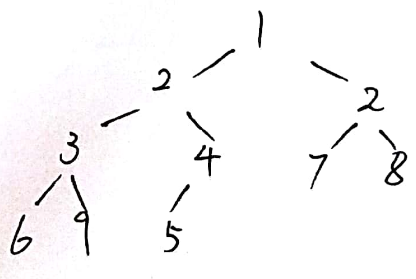
Node at index 2

③



Node at index 1

④



Finished!