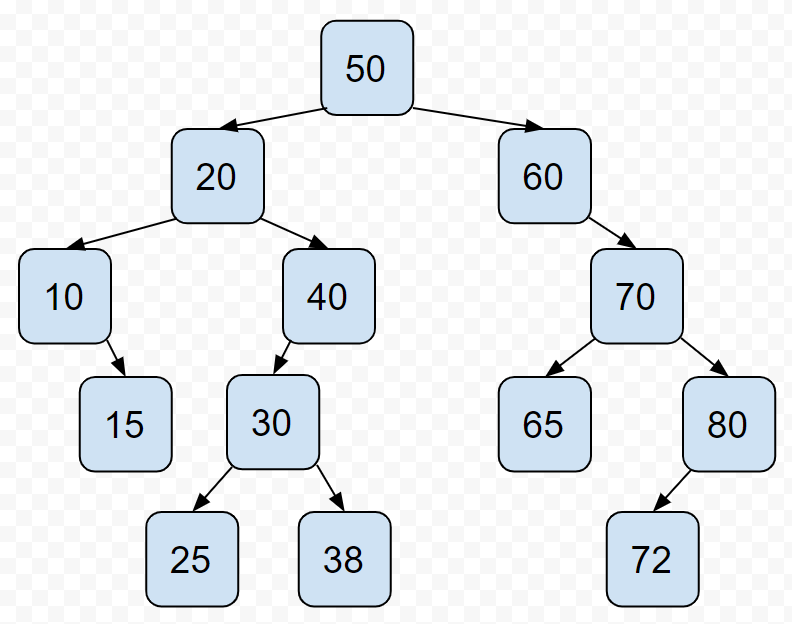
Harrison Cassar

505114980

Dis 1C

TA: Hackett, T.R.

Homework #5

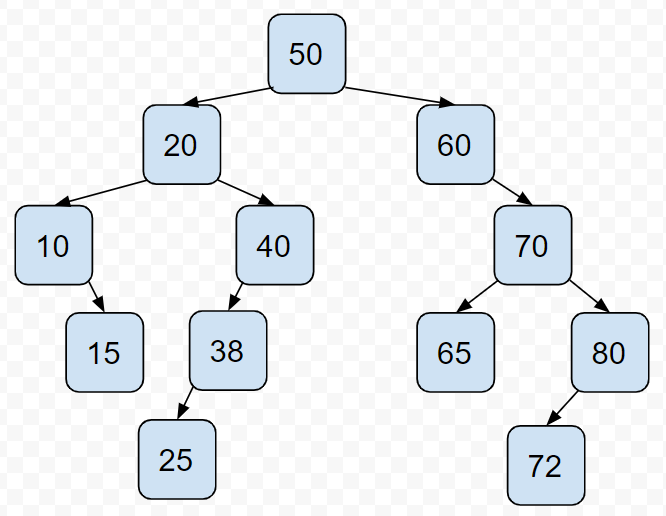
1. A) 

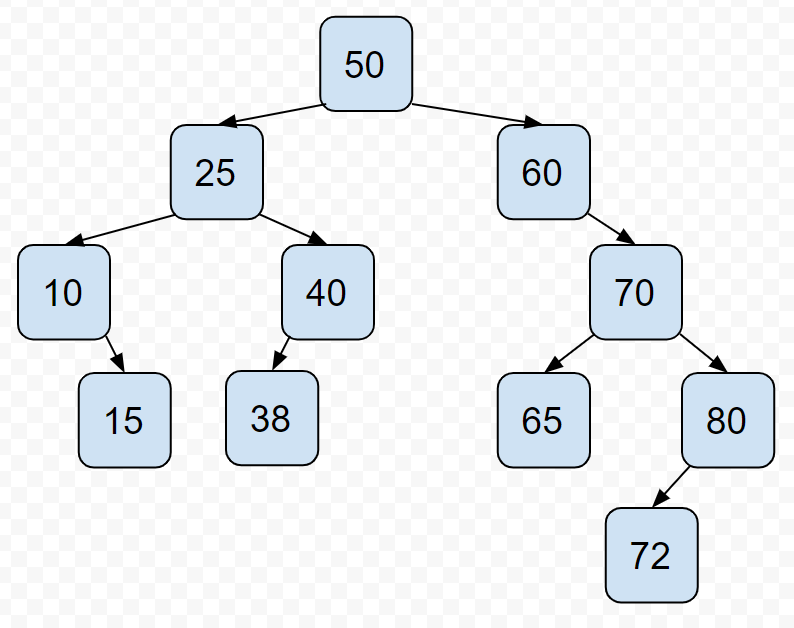
B) In-order: 10, 15, 20, 25, 30, 38, 40, 50, 60, 65, 70, 72, 80

Pre-order: 50, 20, 10, 15, 40, 30, 25, 38, 60, 70, 65, 80, 72

Post-order: 15, 10, 25, 38, 30, 40, 20, 65, 72, 80, 70, 60, 50

C) Two pictures following: (first one depicts the tree after deleting node 30, and the second node depicts the tree after deleting node 20)





2) A)

struct Node

{

int data;

Node\* right;

Node\* left;

Node\* parent;

};

B)

-if root pointer is nullptr (empty tree)

-allocate new node

-point root pointer to new node

-set “left”, “right”, and “parent” pointers to nullptr

-return

-initalize “current” and “parent” pointer to root pointer and nullptr, respectively

-repeating:

-compare value to be inserted to current Node’s value

-if greater:

-set temporary “parent” pointer to “current” pointer

-set temporary “current” pointer to current Node’s right pointer

-else, set temporary “current” pointer to current Node’s left pointer

-if “current” pointer is nullptr, break

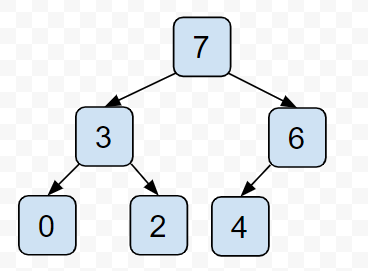
-allocate memory for new node

-set its “data” to inputted value

-set its “right” and “left” Node pointers to nullptr

-set its “parent” pointer to the temporary “parent” pointer

3) A)



B) [7,3,6,0,2,4]

C) [6,3,4,0,2]

4)

A) O(C+S)

B) O(logC + S)

C) O(logC + logS)

D) O(logS)

E) O(1)

F) O(logC + S)

G) O(S\*logS)

H) O(C\*logS)