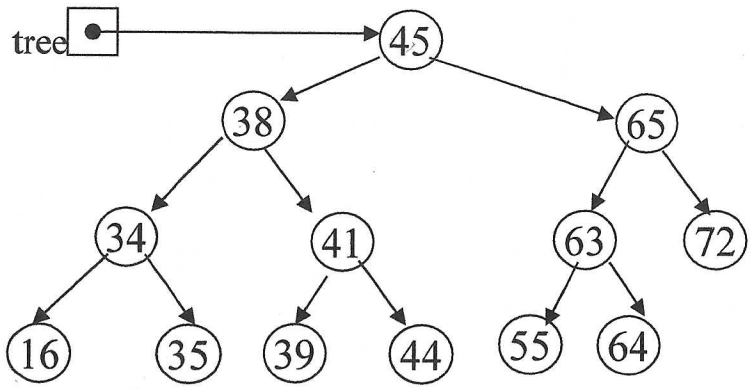
Susan Searles
CMSC 204, Dr. Alexander
Due: October 21, 2020

1. Given the following binary tree:



LNR (a) What is the inorder traversal of the tree?

16 → 34 → 35 → 38 → 39 → 41 → 44 → 45 → 55 → 63 → 64 → 65 → 72

NLR (b) What is the preorder traversal of the tree?

45 → 38 → 34 → 16 → 35 → 41 → 39 → 44 → 65 → 63 → 55 → 64 → 72

LRN (c) What is the postorder traversal of the tree?

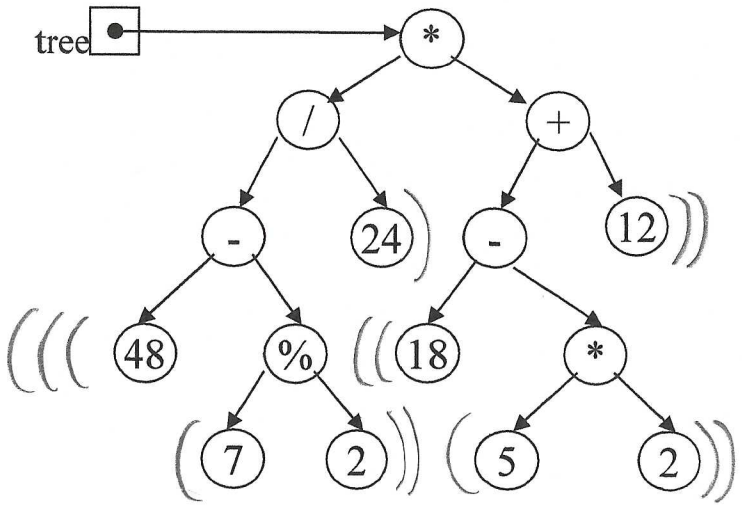
16 → 35 → 34 → 39 → 44 → 41 → 38 → 55 → 64 → 63 → 72 → 65 → 45

(d) What is the height of the tree? What nodes are on level 2?

height of the tree: 4

nodes on level 2: Carrano: 38, 65 (if you consider the root level to be 1)
McAllister: 34, 41, 63, 72 (if you consider the root level to be 0)

2. Given the following binary expression tree:



anything not a leaf node is an operator

postorder traversal of tree

inorder traversal = algebraic form

Post order trav. = same thing

(a) What is the inorder traversal of the tree? LNR

(((48 - (7 % 2)) / 24) * ((18 - (5 * 2)) + 12))

((48 - (7 % 2))/24) * ((18 - (5 * 2)) + 12)

((48 - 1)/24) * ((18 - 10) + 12)

(47/24) * (8 + 12)

LRN (postfix)

(b) What is the postorder traversal of the tree? LRN

48 7 2 % - 24 / 18 5 2 * - 12 + *

(c) What does it evaluate to if using integer division?

20

integer division

(47/24) * (8+12)

1 * 20

(20)

(d) What does it evaluate to if using float division?

39.16̄

float division

(47/24) * (8+12)

1.958̄3 * 20

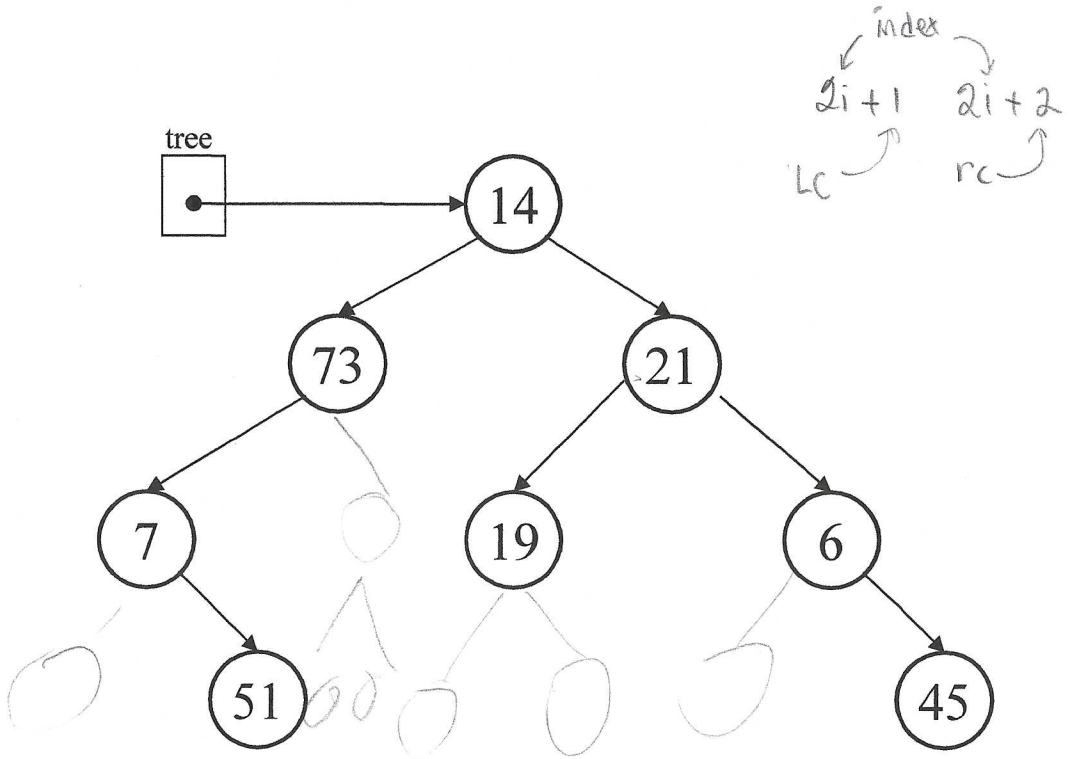
(39.16̄)

3. The elements in a binary tree area to be stored in an array. Each element is a nonnegative int value.

a. What value can you use as a dummy value, if the binary tree is not complete? null

b. Show the contents of the array, given the tree illustrated below

[0]	14	
[1]	73	2(0)+1
[2]	21	2(0)+2
[3]	7	2(1)+1
[4]	null	2(1)+2
[5]	19	2(2)+1
[6]	6	2(2)+2
[7]	null	2(3)+1
[8]	51	2(3)+2
[9]	null	2(4)+1
[10]	null	2(4)+2
[11]	null	2(5)+1
[12]	null	2(5)+2
[13]	null	2(6)+1
[14]	45	2(6)+2



4. Given the array pictured below, draw the binary tree that can be created from its elements.

[0]	35	root
[1]	20	2(0)+1
[2]	71	2(0)+2
[3]	40	2(1)+1
[4]	52	2(1)+2
[5]	63	2(2)+1
[6]	null	2(2)+2
[7]	17	2(3)+1
[8]	25	2(3)+2
[9]	null	2(4)+1
[10]	7	2(4)+2
[11]	null	2(5)+1
[12]	45	2(5)+2