

CSCE 221

Problem Set 9

Jacob Purcell, Texas A&M, Student

I.

A binary tree is defined such that each parent node has no more than 2 children. Let us consider the case of only one node(N_r) with 2 children, the total number of nodes is given by $N_r + 2 = 3nodes$ with a height of 1. At height 2, the expression becomes $N_r(h = 0) + 2(h = 1) + 2 * 2(h = 2)$ which leads to

$$\sum_{h=0}^H 2^h \quad (1)$$

Finding the closed form for this summation from the geometric series,

$$\sum_{n=1}^N ar^n = a(1 + r + r^2 \dots r^N) = S \quad (2)$$

Using the property that $1 + r + r^2 \dots r^N = \frac{r^{N+1} - 1}{r - 1}$,

$$S = a \frac{r^{N+1} - 1}{r - 1} \quad (3)$$

Finally, substituting known values established in eq.1 into eq.3, $r = 2$, $n = h + 1$, $a = 1$ leads to

$$S = 1 \cdot \frac{2^{h+1} - 1}{2 - 1} \quad (4)$$

$$S = 2^{h+1} - 1 \quad (5)$$

II.

Prefix, infix, and postfix notations are expressed as

$$Prefix : [operator][operand][operand] \quad (6)$$

$$Infix : [operand][operator][operand] \quad (7)$$

$$Postfix : [operand][operand][operator] \quad (8)$$

Using PEMDAS and starting from the bottom of the tree to connect the order of operands, the expression will begin as

$$\{(a + b) * c\} - \{d + e\} \quad (9)$$

Using the predefined notations as a map, the expression becomes

$$Prefix : - \{*(+ab)c\}\{+de\} \\ - \{* + abc\}\{+de\} \\ - * + abc + de \quad (10)$$

$$Infix : \boxed{\{(a + b) * c\} - \{d + e\}} \quad (11)$$

$$Postfix : \{(ab+)c*\}\{de+\}- \\ \{ab + c*\}\{de+\}- \\ \boxed{ab + c * de + -} \quad (12)$$

III.

Insertion into a binary tree follows the pattern left \downarrow parent, while right \downarrow parent. Following this pattern and starting with 7, Fig. 1 shows the result of loading a binary tree with the given sequence of numbers.

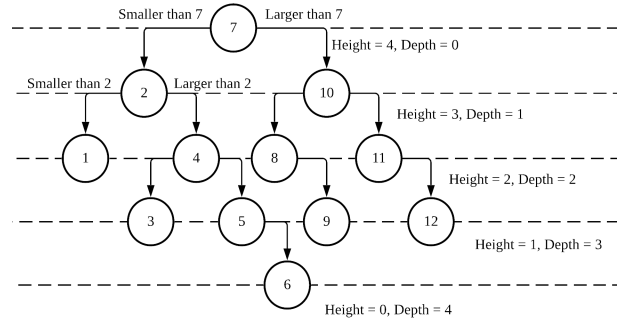


Fig. 1. Each dotted line represents a layer of the tree. Each node in a layer shares the same height and depth. Arrows point to left and right children.

A.

I: 7 is the head node. Since 7 is inserted into an empty tree, it is the only node and all other nodes will be its children.

II: The leaves are 1, 3, 5, 6, 9, and 12 since they have no children.

III: Siblings are child nodes that have the same parent. Since this is a binary tree, siblings come in pairs. Sibling pairs are

2&10

1&4

8&11

3&5

III: Depth is measured from the top starting at 0, the nodes with corresponding depth are

0 : 7
 1 : 2, 10
 2 : 1, 4, 8, 11
 3 : 3, 5, 9, 12
 4 : 6

V: Height is measured from the bottom starting at 0, the nodes with corresponding height are

0 : 6
 1 : 3, 5, 9, 12
 2 : 1, 4, 8, 11
 3 : 2, 10
 4 : 7

B.

Deleting the head node would require some restructuring to maintain tree integrity. We would need to replace it with the next largest value, so we traverse the left side of the right child to the end, then that node is deleted and takes the place of the root node.

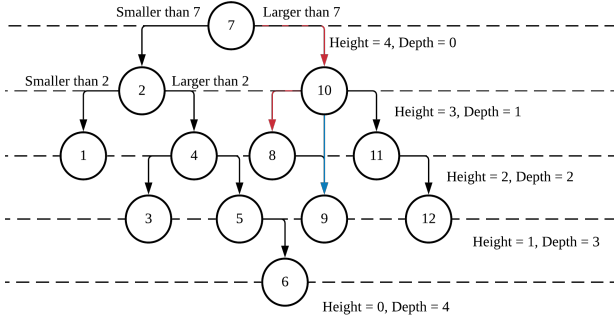


Fig. 2. Red shows the search path until the replacement node is found. Blue shows the creation of a new left child path.

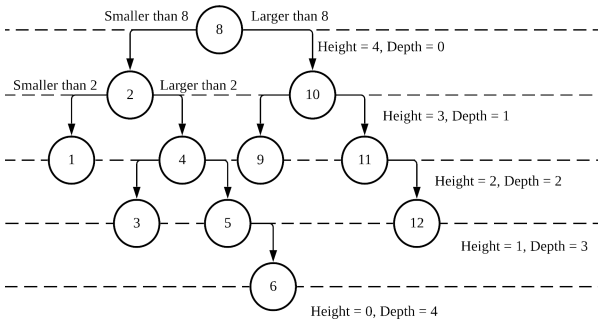


Fig. 3. The final result of deleting the head node.

IV.

A.

Included in "tree_functions.h" lines 108-116. Return value 12 corresponds to Fig. 1.

B.

Included in "tree_functions.h" lines 121-130. Return value 5 corresponds to Fig. 1.

C.

Included in "tree_functions.h" lines 135-144. Return value 4 corresponds to Fig. 1.

"mainps9.cpp" is included as a demonstration where accuracy is confirmed.

Since every function accesses every node once, each function is $O(N)$ although they are defined using recursion.