BANGLADESH UNIVERSITY OF ENGINEERING AND TECHNOLOGY

Department of Computer Science and Engineering

January 2020 CSE 208 Offline Assignment on Advanced Data Structures-I

Red-Black Tree

In this assignment you will have to implement the **Red-Black Tree** data structure. You will read the input from a text file. Each line of the input file will specify one of the following operations: Insert (`I`), Delete (`D`), or Find (`F`), followed by an integer specifying the key the operation will work on. After insertion or deletion,[1] you will have to print the tree (both key and color of each node) in nested parenthesis format.[2] For example, the sample binary search tree in Figure 1 will be written as **8(3(1)(6(4)(7)))(10()(14(13)()))**. For Find operation, print only `True` or `False` based on the query and state of the tree.

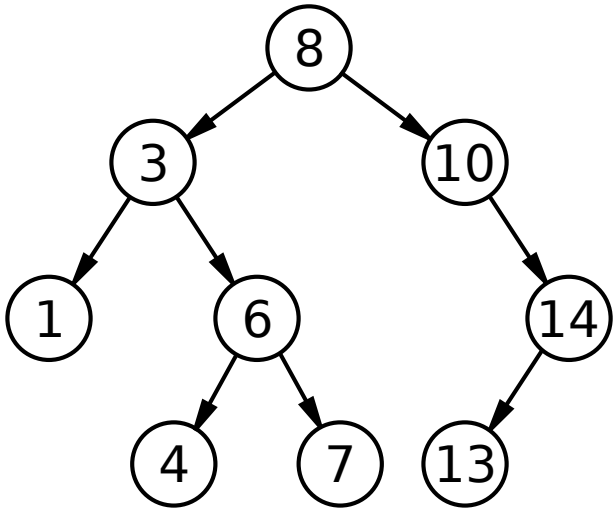

Figure 1: Sample Binary Search Tree

A sample input sequence and its corresponding output is given in table 1.

While you are encouraged to talk to your peers, ask help from teachers, and search relevant resources from online, under no circumstances should you copy code from any source. If found out, you will receive full 100% penalty.

[1] You may assume you will not be given a delete operation on a node that is not present in the tree.

[2] Root node followed by the left sub-tree and right sub-tree encapsulated in parentheses.

Input	Output
F 1	False
I 1	1:b
I 3	1:b()(3:r)
I 2	2:b(1:r)(3:r)
I 5	2:b(1:b)(3:b()(5:r))
I 7	2:b(1:b)(5:b(3:r)(7:r))
I 4	2:b(1:b)(5:r(3:b()(4:r))(7:b))
I 6	2:b(1:b)(5:r(3:b()(4:r))(7:b(6:r)()))
I 8	2:b(1:b)(5:r(3:b()(4:r))(7:b(6:r)(8:r)))
I 10	5:b(2:r(1:b)(3:b()(4:r)))(7:r(6:b)(8:b()(10:r)))
D 7	5:b(2:r(1:b)(3:b()(4:r)))(8:r(6:b)(10:b))
D 1	5:b(3:r(2:b)(4:b))(8:r(6:b)(10:b))
D 8	5:b(3:r(2:b)(4:b))(6:b()(10:r))
F 2	True

Table 1: A sample input and corresponding output