Data Structures and Algorithms

(a) Briefly explain what a binary search tree (BST) is, listing its properties. Is the following binary tree a BST or not, and why?

(b) Describe an optimally efficient algorithm to find the predecessor of a given node $n$ in a BST and explain why it works.

(c) Describe an optimally efficient algorithm for deleting a node $d$ from a BST when neither of $d$’s subtrees is empty. Explain why it works and prove that what remains is still a BST.

(d) Assume that node $l$, whose key is $k_l$, is a leaf of a BST and that its parent is node $p$, with key $k_p$. Prove that, of all the keys in the BST, $k_p$ is either the smallest key greater than $k_l$ or the largest key smaller than $k_l$. 