Weekly Assignment 2

Munia Humaira
Student ID: 21116435
Algorithm Design and Analysis

September 18, 2024

Solution 1:

To prove that, if x is a node in a binary search tree and x has a left child, then the predecessor of x exists in the tree and is located in the left subtree of x, we can use the binary search tree (BST) properties and logical reasoning.
Given, x is a node in a BST and has a left child. We know, the predecessor is the maximum value of the keys in the left subtree.

- Let's denote the left child as y, according to BST property for node x, the keys in the left subtree must follow: $key[y] \leq key[x]$.
- Since, x has a left child y, it must have at least one key smaller than $key[x]$ in the left subtree.
- We know, the predecessor is the maximum value of the keys in the left subtree. Therefore, starting from y we must go as far right as possible.
- If there exists another right child z in the left subtree, then following the right pointer such that $z = max(Tree_{left}) = \emptyset$ i.e., z has no right child, we can say we have found the predecessor of x.

Hence, it is proved that if x has a left child, there exists a predecessor.