

Binary Search Trees (Assignment 6 (CS142))

In this assignment we will build a dynamic data structure that handles insertion and deletion of elements. The data structure for a node in the binary search tree should contain the value at that node and links for two child pointers. Write the functions for insertion, deletion and search in this data structure.

Part 1: We will modify the **search** operation in this data structure. Recall that in a binary search tree, when we search for an element, either we find that element in the data structure or we report that the element is not present in the data structure. We will modify the search operation as follows.

Modified Search (T, key): If an element with value **key** is present in the data structure T , we print ‘The value **key** exists in the data structure’. Else, we find the element in the data structure having value closest to **key**. For this, first we find two elements with values close to **key**, one with value smaller than key and the other with value larger than key. For example, let the data structure contain elements 2, 5, 9. When we search for 6, first it finds 5 and 9, and returns 5 as the closest element. Suppose we insert 6.5 in the data structure. When the data structure is queried with **key** 6, it returns 6.5.

Part 2: Given two numbers a and b , find and return all elements in the data structure with value x such that $a \leq x \leq b$.

Expected input/output behaviour: You will be given as input the values of the keys in an arbitrary order. Assume that all input values are distinct. Initially, the tree must be empty and the tree should be updated with insert, delete and search operations.

Submission: Name your submission as RollNo-Assign6.py and upload on Google classroom.