

316 Data Structures

Assignment #3

(Due: March 31, 2015)

Objective: To practice the programming skills with linked-list-based implementation for binary search trees.

First, you need to implement a Binary Search Tree (BST) using a Linked List.

Then, you provide the following functions in your program. Please write a main.cpp to only include calling functions. Your program must be compiled and tested without any errors. If your program cannot be compiled, you lose **50%** of the total mark.

(Note: **the keys are the data values held by nodes.**)

1. (10 points) Write a function that searches a node in a BST.
2. (10 points) Write a function that inserts a new node into a BST.
3. (10 points) Write a function that deletes an existing node in a BST.
4. (10 points) Write a function that prints out the keys for a BST in sorted order from highest to lowest.
5. (20 points) Write a function to find the depth of a given node in a BST. (**The depth of a node is the length of the path from the root to the node.**)
6. (20 points) The cost of a path in a tree is sum of the keys of the nodes participating in that path. Write a function that returns the cost of the most expensive path from the root to a leaf node in a BST.
7. (20 points) A BST is said to be "balanced" if both of its subtrees are balanced and the height of its left subtree differs from the height of its right subtree by at most 1. Write a function to determine whether a given BST is balanced.

Submission: Put all files related to the assignment to a directory named **A#-FN-LN** (where A# the assignment number, FN your first name and LN your last name). And zip the directory. Then, submit your assignment using SpringBoard.
