

Playing with BinarySearchTree

You have been provided with an implementation of a **Binary Search Tree (BST)**. The following functions are already implemented in the file:

- ***insertItem***: Inserts a new item in the binary search tree.
- ***searchItem***: Searches for an item in the tree.
- ***PrintInOrder***: Prints the tree by an in-order traversal of the tree.
- ***calcHeight/calcNodeHeight***: Calculates the height of an item/node.

For this homework, you are required to add the following functions to the above implementation:

Task 1: Add *getSize* function. Add a new function ***getSize***. This function returns then size of the tree. The size of a tree is the number of nodes in the tree. You should write a recursive function for this task.

Task 2: Add *getMinItem/getMaxItem* function. Add two functions that will find and return the minimum and maximum item of the tree.

Task 3: Add *deleteItem* function. Add a new function ***deleteItem***. The function will delete an existing item from the tree. You must re-structure the tree after deletion according to the idea given in class. In case of deletion of a node where both of its children exist, you must choose the next largest node in the tree for replacement.

Task 4: Add *rangeSearch* function. Add a new function ***rangeSearch***. This function receives three parameters: ***node***, ***leftBound***, ***rightBound***. The last two parameters indicate two integer values. The function should return the number of items in the tree that are greater than or equal to ***leftBound*** and less than or equal to ***rightBound***. **You should write a recursive function for this task.**

You must also satisfy the following requirements:

- You must extend the given code.
- You cannot use any function of C library except ***malloc*** and input output functions.
- You cannot use object oriented programming.
- You must ***free*** unused memory where it is required.