Programming Assignment 4

Due by 2018년 10월 30일 저녁 9시

1. Implement `add()`, `delete()`, `print_inorder()`, `height()` and submit "backend-bst.c". DO NOT CHANGE ANYTHING ELSE.

2. For this programming assignment, we implement the address book with a binary search tree. As always, `data` is the pointer to the binary search tree that stores the data for the address book.

3. Note that we have a new command "`H`" and "`h`" that prints the height of the BST for the Address Book.

4. All the names are supposed to be distinct; `add(`*key*`)` does not add *key* if it is already in the binary search tree and prints out the error message "`Can't add. The name already exists!`".

5. We implement our own memory management almost the same way as in HW3. The only difference is that the nodes have a different structure and size; a node has two links, and we use the right link to make a linked list of nodes when we initialize it. Note that `POOL_SIZE` is defined to be 10 to allow a reasonably sized binary search tree. One important change from HW3 is that `new_node()` returns `NULL` when the pool is empty, instead of just quitting the program. So, it is your responsibilty to check for `NULL` when you call it in `add()`.

6. The function `delete()` is the most difficult part of this assignment. There can be many ways to implement it, but try to stick to what we discussed in the class.