Câu hỏi 5

Chính xác

Chấm điểm của 2,00

In this question, you have to search and print inorder on **AVL tree**. You have o implement functions: **search** and **printInorder** to complete the task. Note that:

- When the tree is null, don't print anything.

- There's a whitespace at the end when print the tree inorder in case the tree is not null.

- When tree contains value, search return true.

```
#include <iostream>
#include <queue>
using namespace std;
#define SEPARATOR "#<ab@17943918#@>#"

enum BalanceValue
{
    LH = -1,
    EH = 0,
    RH = 1
};

template<class T>
class AVLTree
{
public:
    class Node;
private:
    Node *root;
public:
    AVLTree() : root(nullptr) {}
    ~AVLTree(){}

    void printInorder(){
        //TODO
    }

    bool search(const T &value){
        //TODO
    }

    class Node
    {
    private:
        T data;
        Node *pLeft, *pRight;
        BalanceValue balance;
        friend class AVLTree<T>;

    public:
        Node(T value) : data(value), pLeft(NULL), pRight(NULL), balance(EH) {}
        ~Node() {}
    };
};
```

For example:

Test	Result

AVLTree<int> avl;
int arr[] = {10,52,98,32,68,92,40,13,42,63,99,100};
for (int i = 0; i < 12; i++){
avl.insert(arr[i]);
}
avl.printInorder();
cout << endl;
cout << avl.search(10);
	10 13 32 40 42 52 63 68 92 98 99 100
	1

Answer: (penalty regime: 0 %)

```
void printInorderRec(Node* pNode){
        if(!pNode) return;

        printInorderRec(pNode->pLeft);
        cout << pNode->data<< " ";
        printInorderRec(pNode->pRight);
    }

    void printInorder(){
        printInorderRec(this->root);
    }

    bool search(const T &value){
        return searchRec(this->root, value);
    }

    bool searchRec(Node* pNode, T i){
        if (!pNode) return NULL;
        else if (pNode->data == i) return true;
        else if (pNode->data > i) return searchRec(pNode->pLeft, i);
        else return searchRec(pNode->pRight, i);
    }

```

Precheck Kiểm tra

	Test	Expected	Got	

✔	AVLTree<int> avl;
	int arr[] =
	{10,52,98,32,68,92,40,13,42,63,99,100};
	for (int i = 0; i < 12; i++){
	\tavl.insert(arr[i]);
	}
	avl.printInorder();
	cout << endl;
	cout << avl.search(10);
		10 13 32 40 42 52 63 68 92 98
		99 100
		1
			10 13 32 40 42 52 63 68 92 98
			99 100
			1
				✔

Passed all tests! ✔