#include "binarySTree.h"

#include <iostream>

BinarySearchTree::BinarySearchTree() : root(nullptr) {}

Node\* BinarySearchTree::insertRec(Node\* node, int val) {

if (node == nullptr) {

return new Node(val);

}

if (val < node->data) {

node->leftNext = insertRec(node->leftNext, val);

} else {

node->rightNext = insertRec(node->rightNext, val);

}

return node;

}

void BinarySearchTree::insert(int val) {

root = insertRec(root, val);

}

void BinarySearchTree::preOrderRec(Node\* node) {

if (node != nullptr) {

std::cout << node->data << " ";

preOrderRec(node->leftNext);

preOrderRec(node->rightNext);

}

}

void BinarySearchTree::preOrder() {

preOrderRec(root);

std::cout << std::endl;

}

void BinarySearchTree::inOrderRec(Node\* node) {

if (node != nullptr) {

inOrderRec(node->leftNext);

std::cout << node->data << " ";

inOrderRec(node->rightNext);

}

}

void BinarySearchTree::inOrder() {

inOrderRec(root);

std::cout << std::endl;

}

void BinarySearchTree::postOrderRec(Node\* node) {

if (node != nullptr) {

postOrderRec(node->leftNext);

postOrderRec(node->rightNext);

std::cout << node->data << " ";

}

}

void BinarySearchTree::postOrder() {

postOrderRec(root);

std::cout << std::endl;

}