```

// COS30008, Problem Set 4, Problem 2, 2022

#pragma once

#include "BinaryTreeNode.h"

#include <stdexcept>

// Problem 3 requirement
template<typename T>
class BinarySearchTreeIterator;

template<typename T>
class BinarySearchTree
{
private:

    using BNode = BinaryTreeNode<T>;
    using BTreeNode = BNode*;

    BTreeNode fRoot;

public:

    BinarySearchTree()
    {
        fRoot = &BNode::NIL;
    }


    ~BinarySearchTree() {
        if (!empty())
        {
            if (fRoot->left != &BNode::NIL)
            {
                delete fRoot->left;
            }
            if (fRoot->right != &BNode::NIL)
            {
                delete fRoot->right;
            }
        }
    }

    bool empty() const
    {
        return fRoot->empty();
    }

    size_t height() const
    {
        if (fRoot == NULL)
        {
            return 0;
        }
        return fRoot->height();
    }

    bool insert(const T& aKey)
    {
        if (empty()) {
            fRoot = new BNode(aKey);
            return true;
        }
        return fRoot->insert(aKey);
    }

    bool remove(const T& aKey) {

      //return fRoot->remove(aKey, &BTreeNode::NIL);
        //return fRoot->remove(aKey, fRoot->NIL);
        return false;

    }

    // Problem 3 methods

    using Iterator = BinarySearchTreeIterator<T>;

    // Allow iterator to access private member variables
    friend class BinarySearchTreeIterator<T>;

    Iterator begin() const
    {
        return BinarySearchTree<T>(*this);
    }
    Iterator end() const
    {
        return begin().end();
    }
};

```