//steven guo

//11/20/19

// this program buids t bsts by inserting n random keys into an initially empty tree, and then finds the tree height for n=100,500,1000; and t=5,10,15

#include <iostream>

#include <cstdlib>

#include <ctime>

using namespace std;

struct Node {

int data;

Node\* left;

Node\* right;

};

class BST {

private:

Node\* root;

public:

BST() {

root = NULL;

}

void add(int val) {

Node\* newNode = new Node;

newNode->data = val;

newNode->left = NULL;

newNode->right = NULL;

if (root == NULL) {

root = newNode;

}

else {

add(root, newNode);

}

}

void add(Node\*& root, Node\* temp) {

if (root == NULL) {

root = temp;

}

else if (root->data > temp->data) {

add(root->left, temp);

}

else {

add(root->right, temp);

}

}

int height() {

return height(root);

}

int height(Node\* temp) {

if (temp == NULL) {

return -1;

}

else {

int lH = height(temp->left);

int rH = height(temp->right);

return 1 + (lH > rH ? lH : rH);

}

}

};

int main() {

srand(time(NULL));

int N[] = { 100, 500, 1000 };

for (int i = 0; i < 3; ++i) {

for (int t = 5; t <= 15; t += 5) {

double height = 0;

int num = N[i];

for (int nt = 0; nt < t; ++nt) {

BST bst;

for (int j = 0; j < num; ++j) {

bst.add(rand() % 10000);

}

height += bst.height();

}

cout << "Average heigth of a tree of t = " << t << " and N = " << num << " is " << height / t << endl;

}

}

return 0;

}

