A book consists of chapters, chapters consist of sections and sections consist of subsections. Construct a tree and print the nodes. Find the time and space requirements of your method.

#include <iostream>

#include <vector>

#include <string>

using namespace std;

// Node class representing book/chapter/section/subsection

class TreeNode {

public:

string name;

vector<TreeNode\*> children;

TreeNode(string name) {

this->name = name;

}

// Add child to this node

void addChild(TreeNode\* child) {

children.push\_back(child);

}

};

// Recursive function to print tree

void printTree(TreeNode\* root, int depth = 0) {

if (!root) return;

for (int i = 0; i < depth; i++) cout << " ";

cout << "- " << root->name << endl;

for (TreeNode\* child : root->children) {

printTree(child, depth + 1);

}

}

int main() {

// Creating root book

TreeNode\* book = new TreeNode("Book: Data Structures");

// Chapters

TreeNode\* chapter1 = new TreeNode("Chapter 1: Introduction");

TreeNode\* chapter2 = new TreeNode("Chapter 2: Trees");

// Sections in Chapter 1

TreeNode\* section1 = new TreeNode("Section 1.1: What is DS?");

TreeNode\* section2 = new TreeNode("Section 1.2: Importance of DS");

// Sections in Chapter 2

TreeNode\* section3 = new TreeNode("Section 2.1: Binary Trees");

TreeNode\* section4 = new TreeNode("Section 2.2: General Trees");

// Subsections

TreeNode\* subsec1 = new TreeNode("Subsection 2.1.1: Properties");

TreeNode\* subsec2 = new TreeNode("Subsection 2.1.2: Traversals");

// Build the tree

book->addChild(chapter1);

book->addChild(chapter2);

chapter1->addChild(section1);

chapter1->addChild(section2);

chapter2->addChild(section3);

chapter2->addChild(section4);

section3->addChild(subsec1);

section3->addChild(subsec2);

// Print the tree

printTree(book);

return 0;

}

OUTPUT

- Book: Data Structures

- Chapter 1: Introduction

- Section 1.1: What is DS?

- Section 1.2: Importance of DS

- Chapter 2: Trees

- Section 2.1: Binary Trees

- Subsection 2.1.1: Properties

- Subsection 2.1.2: Traversals

- Section 2.2: General Trees