**Bahria University, Lahore Campus**

Department of Computer Science

**Assignment 03**

**(Spring 2023)**

|  |  |  |
| --- | --- | --- |
| Course: | **Data Structures and Algorithm** | Date: \_14-06-2023\_\_\_ |
| Course Code: | CSC-221 | Max Marks: **10** |
| Faculty’s Name: | Ms. Zupash Awais | **Deadline: 14th June 2023(11: 59 PM)** |

Name: AFFAN \_AHMAD\_\_\_ Enroll No: 03-134221-003\_\_ Section: \_BS(cs\_\_\_\_

Instructions:

1. Understanding each question is a part of the assignment.
2. This is a Group Assignment. You need to make group of 2 members each. No more than 2 members allowed. You can perform the task alone if not comfortable in group.
3. Solve the assignment with full honesty. Leave the question that is not clear to you or you are not able to solve, we will discuss it in the class for better understanding.
4. Try to think of a solution of your own. If you will keep on searching for solutions on Google, it will not make your problem-solving ability strong. This assignment is to sharpen your mind a bit or make you able to solve/think of a solution for any problem.
5. Copied assignments (from each other) will be awarded ZERO.
6. *The assignment is to be submitted in the soft form on LMS till the deadline mentioned above.*
7. No need to take a colored print or bring the assignment in files. Submit it as a normal black and white stapled paper

***\*\*Provide the solution in the box for each question\*\****

**Question 1 Marks: 10**

Design Notepad using Binary Search Tree (BST). Create Files (.txt) and store each file name as bst node data. Create the tree accordingly (atleast 10 files). You can write on the file/read from the file. Perform all BST operation:

1. Insertion
2. Deletion
3. Traversals (inorder, preorder, post order)
4. Search
5. Update

|  |
| --- |
| **Source File** |
| **#include <iostream>**  **#include <fstream>**  **#include <string>**  **using namespace std;**  **struct Node {**  **string data;**  **Node\* left;**  **Node\* right;**  **};**  **class Notepad {**  **private:**  **Node\* root;**  **public:**  **Notepad() {**  **root = nullptr;**  **}**    **Node\* createNode(string fileName) {**  **Node\* newNode = new Node();**  **newNode->data = fileName;**  **newNode->left = nullptr;**  **newNode->right = nullptr;**  **return newNode;**  **}**    **Node\* insertNode(Node\* root, string fileName) {**  **if (root == nullptr) {**  **root = createNode(fileName);**  **} else if (fileName < root->data) {**  **root->left = insertNode(root->left, fileName);**  **} else {**  **root->right = insertNode(root->right, fileName);**  **}**  **return root;**  **}**    **Node\* deleteNode(Node\* root, string fileName) {**  **if (root == nullptr)**  **return root;**  **else if (fileName < root->data)**  **root->left = deleteNode(root->left, fileName);**  **else if (fileName > root->data)**  **root->right = deleteNode(root->right, fileName);**  **else {**    **if (root->left == nullptr) {**  **Node\* temp = root->right;**  **delete root;**  **return temp;**  **} else if (root->right == nullptr) {**  **Node\* temp = root->left;**  **delete root;**  **return temp;**  **}**    **Node\* temp = minValueNode(root->right);**    **root->data = temp->data;**    **root->right = deleteNode(root->right, temp->data);**  **}**  **return root;**  **}**    **Node\* minValueNode(Node\* node) {**  **Node\* current = node;**  **while (current && current->left != nullptr)**  **current = current->left;**  **return current;**  **}**    **void inorderTraversal(Node\* root) {**  **if (root != nullptr) {**  **inorderTraversal(root->left);**  **cout << root->data << endl;**  **inorderTraversal(root->right);**  **}**  **}**    **void preorderTraversal(Node\* root) {**  **if (root != nullptr) {**  **cout << root->data << endl;**  **preorderTraversal(root->left);**  **preorderTraversal(root->right);**  **}**  **}**    **void postorderTraversal(Node\* root) {**  **if (root != nullptr) {**  **postorderTraversal(root->left);**  **postorderTraversal(root->right);**  **cout << root->data << endl;**  **}**  **}**    **bool searchFile(Node\* root, string fileName) {**  **if (root == nullptr)**  **return false;**  **else if (fileName == root->data)**  **return true;**  **else if (fileName < root->data)**  **return searchFile(root->left, fileName);**  **else**  **return searchFile(root->right, fileName);**  **}**    **void updateFile(string fileName) {**  **ifstream file(fileName);**  **if (file.good()) {**  **file.close();**  **ofstream outputFile(fileName);**  **cout << "Enter the new content (Enter EOF to finish):\n";**  **string line;**  **while (getline(cin, line) && line != "EOF") {**  **outputFile << line << endl;**  **}**  **outputFile.close();**  **cout << "File '" << fileName << "' updated successfully!" << endl;**  **} else {**  **cout << "File '" << fileName << "' does not exist!" << endl;**  **}**  **}**      **void menu() {**  **int choice;**  **string fileName;**  **ofstream file;**  **do {**  **cout << "---------------------------" << endl;**  **cout << "Notepad Menu" << endl;**  **cout << "---------------------------" << endl;**  **cout << "1. Insert a file" << endl;**  **cout << "2. Delete a file" << endl;**  **cout << "3. Inorder traversal" << endl;**  **cout << "4. Preorder traversal" << endl;**  **cout << "5. Postorder traversal" << endl;**  **cout << "6. Search for a file" << endl;**  **cout << "7. Update a file" << endl;**    **cout << "8. Exit" << endl;**  **cout << "---------------------------" << endl;**  **cout << "Enter your choice: ";**  **cin >> choice;**  **cout << endl;**  **switch (choice) {**  **case 1:**  **cout << "Enter the file name: ";**  **cin >> fileName;**  **root = insertNode(root, fileName);**  **{**  **ofstream file(fileName); // Create the file**  **cout << "File '" << fileName << "' created successfully!" << endl;**  **}**  **break;**  **case 2:**  **cout << "Enter the file name: ";**  **cin >> fileName;**  **root = deleteNode(root, fileName);**  **if (!remove(fileName.c\_str()))**  **cout << "File '" << fileName << "' deleted successfully!" << endl;**  **else**  **cout << "Failed to delete file '" << fileName << "'!" << endl;**  **break;**  **case 3:**  **cout << "Inorder Traversal:" << endl;**  **inorderTraversal(root);**  **break;**  **case 4:**  **cout << "Preorder Traversal:" << endl;**  **preorderTraversal(root);**  **break;**  **case 5:**  **cout << "Postorder Traversal:" << endl;**  **postorderTraversal(root);**  **break;**  **case 6:**  **cout << "Enter the file name: ";**  **cin >> fileName;**  **if (searchFile(root, fileName))**  **cout << "File '" << fileName << "' found!" << endl;**  **else**  **cout << "File '" << fileName << "' not found!" << endl;**  **break;**  **case 7:**  **cout << "Enter the file name: ";**  **cin >> fileName;**  **updateFile(fileName);**  **break;**    **case 8:**  **cout << "Exiting the program..." << endl;**  **break;**  **default:**  **cout << "Invalid choice! Please enter a valid option." << endl;**  **break;**  **}**  **cout << endl;**  **} while (choice != 8);**  **}**  **};**  **int main() {**  **Notepad notepad;**  **notepad.menu();**  **return 0;**  **}**  **Output :**            **Files :** |