

การบ้าน 3 CS213 Data Structures

กำหนดส่งวันที่ 2 พ.ย. 2567 ก่อนเที่ยงคืน

จุดมุ่งหมาย

- เพื่อทบทวนแนวคิดเกี่ยวกับ Tree และ Binary Search Tree
- เพื่อฝึกหัดการเขียนโค้ดสำหรับ Tree และ Binary Search Tree

คำสั่ง

- โจทย์มี 1 ข้อ ให้นักศึกษาเขียนส่วนของการกำหนดคลาสไว้ในไฟล์ BST.h และส่วนของ การสร้างฟังก์ชันของคลาสไว้ในไฟล์ BST.cpp และแยกฟังก์ชัน main ไว้ในไฟล์ main.cpp
- ให้รวมคำตอบทั้งหมดแล้ว zip รวมกันเป็นไฟล์เดียว แล้วตั้งชื่อไฟล์ด้วยรูปแบบ hw3_รหัสนักศึกษา.zip

ให้นักศึกษาอ่านข้อมูลจากไฟล์ Employees-Ordered.txt โดยข้อมูลจะประกอบไปด้วยข้อมูลพนักงานจำนวน 100 คน ได้แก่ ข้อมูล รหัสพนักงาน (ID) เงินเดือน (salary) อายุ (age) และเก็บข้อมูลเอาไว้ใน Array ชนิด vector<Employee> (รายละเอียดของตัวแปรนี้ถูกกำหนดในโค้ดที่กำหนดให้บางส่วนด้านล่าง)

*หมายเลขพนักงานแต่ละคนจะไม่มีทางซ้ำกัน

จากนั้นให้สร้าง **Index structure ด้วย Binary Search Tree (BST)** โดยใช้รหัสพนักงาน (ID) เป็น Key สำหรับโครงสร้างของ BST นี้ โดยกำหนดโค้ดของ Struct Node ใน BST และ Struct Employee สำหรับเก็บ Record ของพนักงานแต่ละคน เป็นดังต่อไปนี้

```
struct Employee{
    int ID;
    int age;
    int salary;
    Employee(int id, int ag, int srl){
        ID = id; age = ag; salary = srl;
    }
};

struct Node{
    Employee* emp;
    Node* left;
    Node* right;
    Node(Employee* e){
        left = right = NULL;
        emp = e;
    }
};
```

โปรดสังเกต: ** Data ใน Node มีชนิดเป็น Pointer ของ Employee หมายความว่า เป็น Pointer ที่ชี้ไปยังพื้นที่ของข้อมูล Employee **

ให้สร้างคลาส BST ที่สามารถทำงานกับฟังก์ชัน main() ที่กำหนดให้ได้อย่างถูกต้องและมีฟังก์ชันอย่างน้อยต่อไปนี (นักศึกษาอาจจะสร้างฟังก์ชันเพิ่มเติมได้)

นักศึกษาอาจจำเป็นต้อง Include headers อย่างน้อยต่อไปนี้

```
#include <iostream>
#include <vector>
#include <fstream>
#include <algorithm>
```

```

class BST{
    vector<Employee> employees;
    Node* root;
    BST(){ root = NULL; }
    void printEmployee(Employee* emp){
        cout << "ID:"    << emp->ID << ", "
            << "Salary:"  << emp->salary << ", "
            << "Age:"     << emp->age << ", "
            << endl;
    }
    void addNode(Employee* e);
    void read_employees(string filename);
    void build_tree_index();
    void deleteIndex(int ID);
    int getHeight();
    void removeAllIndices();
    void shuffleData(){
        random_shuffle(employees.begin(), employees.end());
        cout << "Shuffling Done." << endl;
    }
    void inOrder();
};

```

Method	Detail
BST()	Method ตัวสร้าง (กำหนดให้แล้ว)
void printEmployee(Employee* emp)	Method สำหรับพิมพ์รายละเอียดข้อมูลพนักงานที่รับค่าเข้ามา (กำหนดให้แล้ว)
void addNode(Employee* e)	Method สำหรับเพิ่ม Node ใหม่เข้าไปใน BST
void read_employees(string filename)	Method สำหรับอ่านไฟล์ข้อมูลพนักงานและทำการ push ข้อมูลลงไปในตัวแปร employees ที่มีชนิด vector<Employee> (ดูนิยามคลาสด้านบน) ตามลำดับ จากไฟล์ที่อ่านขึ้นมา
void build_tree_index()	Method สำหรับสร้าง BST Indices สำหรับข้อมูลพนักงานที่อยู่ในตัวแปร employees ที่ได้ทำการอ่านข้อมูลมาจากฟังก์ชัน read_employees(string) กล่าวคือ นักศึกษาต้องทำการสร้างวัตถุของคลาส Employee และเรียกใช้ Method addNode(Employee*) เพื่อเพิ่ม Node ลงไปใน BST ตามลำดับ
void deleteIndex(int ID)	Method สำหรับลบ Index ของพนักงานที่มีหมายเลขพนักงาน = ID ออกจาก BST โดยพื้นที่บน Memory ของ Node ที่ถูกลบควรต้องถูกเคลียร์ (Free) ไปด้วย

	* การลบ เป็นการลบเพียง Index node เท่านั้นแปลว่าการลบนี้ไม่ ควรมีผลกระทบใด ๆ ต่อตัวแปรที่เก็บข้อมูล employees
<code>int getHeight()</code>	Method ที่คืนค่าความสูงของต้นไม้
<code>void removeAllIndices()</code>	Method ที่ลบค่า Node ทุก Nodes ออกจาก BST หลังจากลบแล้ว ตัวแปร root ควรจะมีค่าเท่ากับ NULL และพื้นที่บน Memory ของ Node ที่ถูกลบทั้งหมดควรต้องถูกเคลียร์ (Free) ไปด้วย
<code>void shuffleData()</code>	Method สำหรับสลับลำดับของข้อมูลพนักงานที่เก็บอยู่ในตัวแปร employees (กำหนดค่าให้แล้ว)
<code>void inOrder()</code>	Method สำหรับท่อง BST แบบ InOrder โดยการ visit node ควร เรียกใช้ Method <code>printEmployee(Employee*)</code> โดย ผลลัพธ์ที่ถูกต้องคือโปรแกรมต้องแสดงผลข้อมูลพนักงานแบบ เรียงลำดับตาม ID (จากน้อยไปหามาก)

โดยกำหนดฟังก์ชัน `main()` ดังต่อไปนี้ (นักศึกษาห้ามเปลี่ยนแปลงสิ่งใดในฟังก์ชันนี้)

```
#include<iostream>
#include<BST.h>
using namespace std;

int main(){

    BST bst;
    bst.read_employee("Employees-Ordered.txt");

    bst.build_tree_index();
    cout << "The height of BST:" << bst.getHeight() << endl;
    bst.removeAllIndices();

    int sN =5;
    cout << "...Try shuffling a number of times ..." << endl;

    for (int i=0;i<sN;i++){
        cout << "#" << i +1 << "shuffle ..." << endl;
        bst.shuffleData();
        bst.build_tree_index();
        cout << "The height of BST:" << bst.getHeight() << endl;
        bst.removeAllIndices();
    }

    bst.build_tree_index();
    bst.inOrder();

    return 0;
}
```

}

ตัวอย่างผลลัพธ์

BST has been constructed.
The height of BST: 99
All indices have been deleted
... Try shuffling a number of times ...
#1 shuffle ...
Shuffling Done.
BST has been constructed.
The height of BST: 13
All indices have been deleted
#2 shuffle ...
Shuffling Done.
BST has been constructed.
The height of BST: 11
All indices have been deleted
#3 shuffle ...
Shuffling Done.
BST has been constructed.
The height of BST: 13
All indices have been deleted
#4 shuffle ...
Shuffling Done.
BST has been constructed.
The height of BST: 12
All indices have been deleted
#5 shuffle ...
Shuffling Done.
BST has been constructed.
The height of BST: 15
All indices have been deleted
BST has been constructed.
ID:6500001, Salary:68134, Age:20,
ID:6500002, Salary:47909, Age:56,
ID:6500003, Salary:48791, Age:26,

ID:6500004, Salary:50278, Age:26,
ID:6500005, Salary:50014, Age:45,
ID:6500006, Salary:59441, Age:55,
ID:6500007, Salary:26711, Age:18,
ID:6500008, Salary:43686, Age:20,
ID:6500009, Salary:26846, Age:30,
ID:6500010, Salary:11318, Age:20,
ID:6500011, Salary:66724, Age:55,
ID:6500012, Salary:46006, Age:41,
ID:6500013, Salary:68651, Age:35,
ID:6500014, Salary:27834, Age:57,
ID:6500015, Salary:62099, Age:55,
ID:6500016, Salary:42786, Age:56,
ID:6500017, Salary:29216, Age:55,
ID:6500018, Salary:12093, Age:50,
ID:6500019, Salary:31175, Age:52,
ID:6500020, Salary:57536, Age:47,
ID:6500021, Salary:39060, Age:23,
ID:6500022, Salary:65867, Age:55,
ID:6500023, Salary:18610, Age:22,
ID:6500024, Salary:17173, Age:52,
ID:6500025, Salary:13503, Age:41,
ID:6500026, Salary:58030, Age:58,
ID:6500027, Salary:66720, Age:18,
ID:6500028, Salary:20148, Age:31,
ID:6500029, Salary:26380, Age:44,
ID:6500030, Salary:30729, Age:59,
ID:6500031, Salary:23656, Age:25,
ID:6500032, Salary:50832, Age:39,
ID:6500033, Salary:46943, Age:50,
ID:6500034, Salary:47772, Age:50,
ID:6500035, Salary:65299, Age:57,
ID:6500036, Salary:61656, Age:46,
ID:6500037, Salary:57614, Age:34,
ID:6500038, Salary:10487, Age:34,
ID:6500039, Salary:42415, Age:57,
ID:6500040, Salary:65995, Age:40,

ID:6500041, Salary:17260, Age:29,
ID:6500042, Salary:39354, Age:48,
ID:6500043, Salary:36511, Age:33,
ID:6500044, Salary:31614, Age:35,
ID:6500045, Salary:57375, Age:59,
ID:6500046, Salary:62885, Age:54,
ID:6500047, Salary:61892, Age:18,
ID:6500048, Salary:42209, Age:23,
ID:6500049, Salary:39354, Age:29,
ID:6500050, Salary:62977, Age:19,
ID:6500051, Salary:44406, Age:50,
ID:6500052, Salary:20155, Age:43,
ID:6500053, Salary:69075, Age:59,
ID:6500054, Salary:47476, Age:20,
ID:6500055, Salary:40165, Age:49,
ID:6500056, Salary:11897, Age:42,
ID:6500057, Salary:13492, Age:23,
ID:6500058, Salary:17281, Age:56,
ID:6500059, Salary:37830, Age:18,
ID:6500060, Salary:45738, Age:57,
ID:6500061, Salary:27942, Age:30,
ID:6500062, Salary:11610, Age:44,
ID:6500063, Salary:36869, Age:44,
ID:6500064, Salary:54900, Age:22,
ID:6500065, Salary:21173, Age:58,
ID:6500066, Salary:64451, Age:36,
ID:6500067, Salary:28489, Age:24,
ID:6500068, Salary:56722, Age:39,
ID:6500069, Salary:16532, Age:42,
ID:6500070, Salary:68942, Age:22,
ID:6500071, Salary:62129, Age:57,
ID:6500072, Salary:29017, Age:34,
ID:6500073, Salary:31877, Age:21,
ID:6500074, Salary:53090, Age:56,
ID:6500075, Salary:65735, Age:44,
ID:6500076, Salary:53055, Age:54,
ID:6500077, Salary:67755, Age:46,

ID:6500078, Salary:24371, Age:38,
ID:6500079, Salary:16903, Age:38,
ID:6500080, Salary:59394, Age:44,
ID:6500081, Salary:36643, Age:44,
ID:6500082, Salary:63954, Age:18,
ID:6500083, Salary:51438, Age:37,
ID:6500084, Salary:45646, Age:32,
ID:6500085, Salary:36699, Age:41,
ID:6500086, Salary:10230, Age:57,
ID:6500087, Salary:18060, Age:60,
ID:6500088, Salary:37414, Age:52,
ID:6500089, Salary:62709, Age:19,
ID:6500090, Salary:47819, Age:20,
ID:6500091, Salary:55318, Age:39,
ID:6500092, Salary:15905, Age:50,
ID:6500093, Salary:59928, Age:40,
ID:6500094, Salary:60905, Age:23,
ID:6500095, Salary:15021, Age:35,
ID:6500096, Salary:62740, Age:28,
ID:6500097, Salary:43203, Age:48,
ID:6500098, Salary:61843, Age:24,
ID:6500099, Salary:65399, Age:51,
ID:6500100, Salary:40745, Age:41,