KIỂM TRA CUỐI KỲ (100')

Thứ 4 ngày 18 tháng 12 năm 2019

Quy định

1 Quy định nộp bài

- Sinh viên thực hiện cài đặt theo đúng tên hàm được khai báo trong đề bài.
- Sinh viên **KHÔNG ĐƯỢC PHÉP THAY ĐỔI** các khai báo hàm trong đề bài.
- Chỉ nộp tập tin *.cpp (*.h nếu có).
- Bài nộp **PHẢI CÓ** ít nhất 1 tập tin tên `main.cpp` chứa hàm `main` thực thi toàn bộ bài làm (các yêu cầu đề bài) của sinh viên.
- Nén các tập tin trên thành tập tin MSSV_treeType.zip và nộp tập tin nén này. Ở đây treeType là loại cây sinh viên dùng để cài đặt, có thể là AVL hoặc BST. Ví dụ 1812001_BST.zip.

2 Quy định chấm bài

- Sinh viên chỉ sử dụng các hàm trong thư viện chuẩn (bài làm sẽ được biên dịch bằng g++).
- KHÔNG chấm ý tưởng, chỉ có đúng hoặc sai.
- Những trường hợp sau đây sẽ bị 0 điểm bài thi:
  - Nộp sai quy định.
  - Bài làm giống nhau.
  - KHÔNG BIÊN DỊCH ĐƯỢC.
  - LẶP VÔ TẬN.
- **Điểm tổng cuối cùng của bài kiểm tra sẽ được tính như sau:**

```
Điểm bài thi = điểm tổng các bài * hệ số
```

Nếu sinh viên cài đặt bằng cây nhị phân tìm kiếm (BST) thì hệ số là 0.8. Nếu sinh viên cài đặt bằng cây nhị phân tìm kiếm cân bằng (AVL) thì hệ số là 1.0.

Nội dung

1 Mô tả dữ liệu

Dữ liệu được dùng trong bài kiểm tra này là dữ liệu về điểm thi của một tỉnh (thông tin thật của học sinh đã được thay đổi).

Tập tin `data.txt` được cung cấp có một phần nội dung như sau:

```
Số Báo Danh,Toán,Ngữ Văn,Ngoại Ngữ,Ghi Chú
LC1200001,4.4,7.0,3.2,N1
LC1200002,4.6,5.5,3.4,N1
LC1200003,5.6,5.25,3.8,N1
LC1200004,7.2,4.25,3.4,N1
LC1200005,7.0,6.25,3.2,N1
LC1200007,6.0,4.75,6.2,N1
LC1200008,3.2,3.75,3.2,N1
LC1200009,5.6,6.25,4.0,N1
LC1200010,5.0,5.0,4.6,N1
LC1200011,4.6,3.25,3.0,N1
LC1200012,7.0,4.75,5.8,N1
LC1200013,3.6,5.0,5.2,N1
LC1200014,7.6,4.25,4.4,N1
```

Trong đó:

- Dòng đầu tiên thể hiện tên các trường thông tin có trong tập tin.
- Những dòng tiếp theo thể hiện thông tin học sinh, mỗi trường thông tin cách nhau bởi 1 dấu phẩy (,).

2 Yêu cầu

Sinh viên thực hiện lưu trữ thông tin từ tập tin `data.txt` bằng cây nhị phân. Sinh viên tự định nghĩa cấu trúc `NODE` sao cho phù hợp nhưng phải có tối thiểu những thông tin sau đây:

- `key` giá trị khóa
- `pLeft` con trỏ đến `NODE` con bên trái
- `pRight` con trỏ đến `NODE` con bên phải

Điểm trung bình được dùng ở các yêu cầu sau đây là điểm trung bình của ba môn có trong thông tin của học sinh.

```
điểm trung bình = (Toán + Ngữ văn + Ngoại ngữ)/3
```

Câu 1: (3 điểm)

Xuất ra màn hình `ID` của học sinh theo thứ tự tăng dần điểm trung bình. Những thí sinh có điểm trung bình bằng nhau sẽ có thứ tự ngẫu nhiên.

- `void DisplayIDs(NODE* pRoot);`
- Input: - `pRoot` - Node gốc của cây nhị phân đang lưu trữ thông tin các học sinh.

Câu 2: (3 điểm)

Tính chiều cao cây nhị phân đang lưu trữ thông tin các học sinh.

- `int Height(NODE* pRoot);`
- Input: - `pRoot` - Node gốc của cây nhị phân đang lưu trữ thông tin các học sinh.
- Output: Chiều cao của cây nhị phân đầu vào, có kiểu dữ liệu là `int`.

Câu 3: (3 điểm)

Tìm `ID` của (những) học sinh có điểm trung bình là `s`.

- `vector<string> SearchEqual(NODE* pRoot, float s);`
- Input: - `pRoot` - Node gốc của cây nhị phân đang lưu trữ thông tin các học sinh.
  - `s` - điểm trung bình muốn tìm kiếm.
- Output: Kiểu dữ liệu `vector<string>` lưu trữ danh sách định danh (ID) của học sinh có điểm trung bình là `s`. Trả về vector rỗng nếu không tìm thấy.

Câu 4: (1 điểm)

Tìm `ID` của (những) học sinh có điểm trung bình lớn hoặc hoặc bằng `s`:

- `vector<string> SearchGreater(NODE* pRoot, float s);`
- Input: - `pRoot` - Node gốc của cây nhị phân đang lưu trữ thông tin các học sinh.
  - `s` - điểm trung bình ngưỡng để so sánh.
- Output: Kiểu dữ liệu `vector<string>` lưu trữ danh sách định danh (ID) của học sinh có điểm trung bình lớn hơn hoặc bằng `s`.

The End