

Final Lab Exam (E)

Thời gian làm bài: 90 phút

1 Quy định nộp bài

- Sinh viên phải thực hiện bài kiểm tra tại phòng máy.
- Sinh viên thực hiện các hàm với Prototype có sẵn, trong một file `MSSV.cpp` duy nhất. (Chỉ cần viết hàm, không cần viết hàm `main()`). Nộp `MSSV.cpp`

Lưu ý: Thay cụm MSSV thành mã số sinh viên của bạn.

2 Quy định chấm bài

- Chấm bài bằng trình biên dịch g++ (thư viện chuẩn)
- Đối với những hàm sinh viên không thực hiện được → giữ nguyên trạng thái ban đầu.
- 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.
 - LẬP VÔ TẬN.

3 Xử lý dữ liệu đầu vào

Cho file `g1.v2.jl` chứa dữ liệu thời gian các chuyến bay của hãng hàng không quốc gia.

```
{from,to,[name of airliner,the number of business seats,the number of economy seats,hours,minutes]}
{"Myanmar,South Sudan": ["Comac ARJ-900/C909-200 90", "11 business, 84 economy", "9 hours, 25 minutes"]}
{"Faeroe Islands,Ethiopia": ["Comac C919", "92 business, 98 economy", "9 hours, 16 minutes"]}
{"Italy,Greece": ["Bombardier C-110 100", "8 business, 117 economy", "1 hour, 36 minutes"]}
{"Netherlands,Tunisia": ["Antonov 24", "21 business, 31 economy", "3 hours, 4 minutes"]}
...
```

Đọc file và chuyển dữ liệu về dạng cấu trúc theo yêu cầu đề bài.

(Sinh viên sang trang kế tiếp để xem đề bài.)

4 Quy ước

Đọc ghi file và hàm todo tương tự như Test 1 & 2. Các câu cách nhau =====.

Sai hàm insert ở (câu đầu của mỗi phần) cây/đồ thị => KHÔNG CHẤM những câu còn lại

Phúc khảo cuối kì: -3 điểm cho MỖI lỗi mã nguồn sai mà giáo viên phải sửa

Quy ước 1 số thông số của cây: $h_{leaf} = 1$, $level_{root} = 0$

data.tx có cấu trúc 2 phần, dòng đầu tiên là dữ liệu dùng để thực thi phần *Đồ thị*, những dòng còn lại là dữ liệu cho *Cây AVL*:

```
Vietnam,10          //country,x (8 < alpha < 23)
a b a               //AVLNode
h c m u s          //AVLNode
i o o i             //AVLNode
a b a b a           //AVLNode
a i a a i a         //AVLNode
g o o d l u c k     //AVLNode
```

5 Đồ thị

Xét đồ thị $G(V, E)$ với tập đỉnh V là tập hợp các quốc gia(nước), E là đường bay giữa 2 quốc gia tương ứng với các đỉnh trong V .

Câu 1 (1 điểm) Từ file dữ liệu *g1.v2.jl*, với các struct được định nghĩa như trong file *func.cpp* và *27127000.cpp*, hãy tìm các chuyến bay của hãng Airbus hoặc Boeing, và xây dựng 1 đồ thị G vô hướng biểu diễn các chuyến bay vừa tìm được. In ra 10 nước bất kì có đường bay trực tiếp đến nước **country**.

Output mẫu:

Vietnam: Cameroon, Benin, Lebanon, DRC, Saudi Arabia, Lithuania, UAE, Afghanistan, Belarus, Namibia,

Câu 2 (2 điểm) Gọi c_degree là bậc của nước **country**. Viết hàm đếm số nước có bậc lớn hơn c_degree và in ra nước có bậc nhỏ nhất lớn hơn c_degree . Nếu có nhiều nước có cùng bậc nhỏ nhất, in ra tất cả các nước này

Output mẫu:

200 vertice, the lowest: Hong Kong, Indonesia

Câu 3 (3 điểm) Gọi C (capacity) là số chuyến bay tối đa mà sân bay của nước **country** có thể cho đáp đất trong mỗi khoảng thời gian 60 phút, từ $[x : 00 - x : 59]$. Giả sử tất cả các chuyến bay đến **country** (bay trực tiếp + bay có quá cảnh) cùng xuất phát lúc 8 : 00, và không có thời gian nghỉ giữa các chuyến bay quá cảnh (từ A bay đến B, rồi ngay lập tức khởi hành đến C).

Hỏi giá trị tối thiểu của C là bao nhiêu, để sân bay của **country** không bị quá tải trong khung giờ từ $[x : 00 - x : 59]$ với x cho trước. Yêu cầu xuất: in số C của **country**

Output mẫu:

capacity = 6

Dưới đây là phần giải thích thêm cho kết quả trên (sinh viên không cần giải thích, chỉ cần trả về kết quả đúng):

['Myanmar', 123], ['Philippines', 161], ['Malaysia', 134], ['Taiwan', 162] (bay trực tiếp tới Vietnam)
 + ['Hong Kong', 35] ['Macao', 108] (bay quá cảnh qua Macao)
 + ['Macao', 35] ['Hong Kong', 35] ['Macao', 108] (Macao -> Hong Kong -> Macao -> Vietnam)

6 Cây AVL

Câu 4 (2 điểm) Cho cấu trúc Node như sau:

```
struct Node{  
    vector<char> A;  
    Node* left; Node* right;  
};
```

Hãy xây dựng một cây AVL bằng cách thêm lần lượt từng Node, với khóa của mỗi Node là mảng các kí tự `vector<char>` được đọc theo dòng từ trên xuống dưới trong file `data.txt`. Xét 2 khóa A và B , định nghĩa của $A > B$ khi thỏa các ưu tiên sau đây:

- Một vector được gọi là palindrome khi giá trị của vector này không đổi khi ta đảo ngược nó.
- Ưu tiên 1: Nếu A là palindrome và B không là palindrome thì A lớn hơn B , hoặc ngược lại.
- Ưu tiên 2: Nếu cả 2 đều là palindrome hoặc cả 2 đều không là palindrome thì vector nào có nhiều số lượng nguyên âm nhiều hơn thì lớn hơn.
- Nếu 2 khóa có cùng cả ưu tiên 1 và 2 thì không thêm khóa mới vào cây.

Duyệt LNR cho cây AVL vừa tạo.

Output mẫu:

```
hcmus goodluck aba ababa iooi aiaaia
```

Câu 5 (2 điểm) Xóa tất cả các Node không phải palindrome và có ít hơn 3 nguyên âm. Duyệt LNR cho cây vừa xóa.

Output mẫu:

```
goodluck aba ababa iooi aiaaia
```