

# KIỂM TRA CUỐI KỲ - 20CTT3

Thời gian: 75'

Mã đề: 2

## Quy định

Sinh viên xem file [Practical Final Test Regulations](#) trên Moodle.

## Nội dung

### 1 Câu 1 (5 điểm)

Cho định nghĩa cấu trúc thể hiện thông tin của một node trong cây AVL như sau:

---

```
struct AVLNode {  
    int p, q;  
    Node* pLeft;  
    Node* pRight;  
};
```

---

Trong đó, p và q lần lượt là chiều dài và chiều rộng của một hình chữ nhật. p và q là số nguyên dương.

Ta quy định hình chữ nhật lớn hơn sẽ có diện tích lớn hơn. Nếu 2 hình chữ nhật có cùng diện tích, hình có chu vi lớn hơn sẽ lớn hơn.

Hãy cài đặt các hàm sau:

- Hàm **Insert** cho phép thêm một node vào cây AVL.  
Prototype: `void Insert(AVLNode* &root, int pInp, int qInp)`
- Hàm **Remove** cho phép xóa một node có giá trị (p, q) bằng với cặp tham số (qInp, pInp) được truyền vào hàm.  
Prototype: `void Remove(AVLNode* &root, int pInp, int qInp)`
- Hàm **CreateTree** cho phép tạo cây AVL từ **mảng** các giá trị (p, q) được truyền vào hàm.  
Prototype: `AVLNode* CreateTree(vector<int> pList, vector<int> qList)`
- Hàm **LevelOrder** cho phép duyệt cây AVL theo mức, giá trị mỗi node in ra có định dạng (p, q).  
Prototype: `void LevelOrder(AVLNode* root)`

Lưu ý: KHÔNG SỬA các prototype (nguyên mẫu) hàm. Thực hiện cài đặt hàm **Bai01()** để mô phỏng hoàn thành bài 1. **Bai01** gọi hàm **CreateTree** để tạo cây, gọi hàm **Remove** xóa 1 node bất kỳ và gọi hàm **LevelOrder** để in ra các giá trị trong cây.

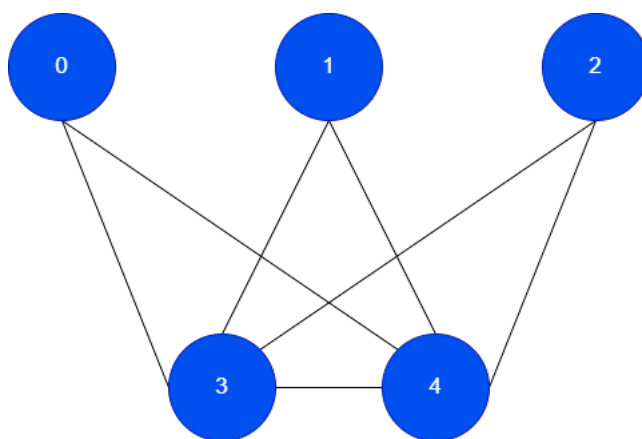
## 2 Câu 2 (5 điểm)

Một đồ thị  $n$  đỉnh được xem là có hình quyển sách được định nghĩa như sau:

- Không có khuyên.
- Có 2 đỉnh, tạo thành 1 cạnh, nối với tất cả các đỉnh còn lại. Cạnh này được xem là gáy sách.
- $n - 2$  đỉnh còn lại, là các trang sách, chỉ nối với các đỉnh của gáy sách.

Hãy cài đặt hàm `bool solution(vector<vector<bool>> adj)` nhận input là một ma trận kề, cho biết đồ thị tương ứng có phải là đồ thị hình quyển sách hay không.

Ví dụ, cho đồ thị hình quyển sách được thể hiện ở Hình 1.



Hình 1: Đồ thị hình quyển sách

Ma trận kề tương ứng cho đồ thị trên như sau:

```
[[false,false,false,true,true],  
[false,false,false,true,true],  
[false,false,false,true,true],  
[true,true,true,false,true],  
[true,true,true,true,false]]
```

Hàm `solution` với tham số là ma trận kề trên sẽ trả về `true`.

Lưu ý: KHÔNG SỬA prototype (nguyên mẫu) hàm `solution`. Sau khi cài đặt xong hàm `solution`, thực hiện cài đặt hàm `Bai02()` để mô phỏng hoàn thành bài 2. Trong `Bai02` in ra kết quả khi gọi hàm `solution`.

— HẾT —