# **Tuần 3**

## **Bài 3: DSC: add\_edge nâng cao**

## **Bài 4: DSC: hàm adjacent**

## **Bài 5: Tổng hợp danh sách cung**

## **Bài 6: Danh sách cung: liệt kê đỉnh kề**

## **Bài 7: Ma trận kề: hàm init\_graph() và add\_edge()**

### **Ex1:**

Cho cấu trúc dữ liệu đồ thị **Graph** được cài đặt bằng phương pháp “**Ma trận kề**” dùng để lưu trữ các **đơn đồ thị vô hướng**. Hãy hoàn chỉnh chương trình bên dưới để đọc đồ thị từ bàn phím và in ra ma trận kề của đồ thị ra màn hình.

Viết mã lệnh của bạn (hàm init\_graph() và hàm add\_edge()) vào chỗ ba chấm (...).

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra.
* Không nộp toàn bộ chương trình, chỉ nộp phần mã lệnh của bạn.

### **Ex2:**

Cho cấu trúc dữ liệu đồ thị **Graph** được cài đặt bằng phương pháp “**Ma trận kề**” dùng để lưu trữ các **đồ thị vô hướng (có thể chứa đa cung và chứa khuyên)**. Hãy hoàn chỉnh chương trình bên dưới để đọc đồ thị từ bàn phím và in ra ma trận kề của đồ thị ra màn hình.

Viết mã lệnh của bạn (hàm init\_graph() và hàm add\_edge()) vào chỗ ba chấm (...).

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra.
* Không nộp toàn bộ chương trình, chỉ nộp phần mã lệnh của bạn.

Ex3:

Cho cấu trúc dữ liệu đồ thị **Graph** được cài đặt bằng phương pháp “**Ma trận kề**” dùng để lưu trữ các **đồ thị có hướng (có thể chứa đa cung và chứa khuyên)**. Hãy hoàn chỉnh chương trình bên dưới để đọc đồ thị từ bàn phím và in ra ma trận kề của đồ thị ra màn hình.

Viết mã lệnh của bạn (hàm init\_graph() và hàm add\_edge()) vào chỗ ba chấm (...).

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra.
* Không nộp toàn bộ chương trình, chỉ nộp phần mã lệnh của bạn.

## **Bài 8: Ma trận kề: hàm add\_edge()**

## **Bài 9 Ma trận kề: In ma trận**

## **Bài 10 Ma trận kề: Hàm degree()**

## **Bài 11 Ma trận kề: Liệt kê đỉnh kề**

Ex1:

Viết chương trình bằng ngôn ngữ C cho phép người dùng nhập dữ liệu của một **đồ thị vô hướng (có thể chứa đa cung và chứa khuyên)** và in các đỉnh kề của các đỉnh ra màn hình.

Biểu diễn đồ thị bằng phương pháp "**Ma trận kề**".

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 2 số nguyên n và m cách nhau một khoảng trắng, n: số đỉnh, m: số cung
* m dòng tiếp theo, mỗi dòng chứa 2 số nguyên u v cách nhau 1 khoảng trắng mô tả cung (u, v).

**Đầu ra**

* In ra n dòng, dòng thứ i in các đỉnh kề của đỉnh i, cách nhau 1 khoảng trắng, theo thứ tự tăng dần (không lặp lại), ví dụ: **neighbours(2) = 1 2 4**

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

Ex2:

Viết chương trình bằng ngôn ngữ C cho phép người dùng nhập dữ liệu của một **đồ thị có hướng (có thể chứa đa cung và khuyên)** và in các đỉnh kề của các đỉnh ra màn hình.

Biểu diễn đồ thị bằng phương pháp "**Ma trận kề**".

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 2 số nguyên n và m cách nhau một khoảng trắng, n: số đỉnh, m: số cung
* m dòng tiếp theo, mỗi dòng chứa 2 số nguyên u v cách nhau 1 khoảng trắng mô tả cung (u, v).

**Đầu ra**

* In ra n dòng, dòng thứ i in các đỉnh kề của đỉnh i, cách nhau 1 khoảng trắng, theo thứ tự tăng dần (không lặp lại), ví dụ: **neighbours(2) = 1 2 4**

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

Ex2:

## **Bài 13,14,15 Chuyển đổi giữa các PP biểu diễn**

### **Ex1:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào ma trận kề của một **đồ thị vô hướng** và in các cung của nó ra màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp theo mô tả ma trận kề của đồ thị. Mỗi dòng có n số nguyên, cách nhau 1 khoảng trắng.

**Đầu ra**

* In các cung của đồ thị vừa nhập ra màn hình, mỗi cung trên 1 dòng theo mẫu: u v .  
    
   (u≤v)  
    
   (u≤v)
* Nếu có 2 cung (u1, v1) và (u2, v2) thì cung nào có u nhỏ sẽ được in ra trước. Nếu u bằng nhau thì cung nào có v nhỏ sẽ được in ra trước.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, đọc ma trận kề và lưu trực tiếp vào G.A
* Duyệt qua các ô của tam giác trên của ma trận kề G.A. Con số trong ô G.A[u][v] cho biết số cung đi từ u đến v.

### **Ex2:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào ma trận kề của một **đồ thị có hướng** và in các cung của nó ra màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp theo mô tả ma trận kề của đồ thị. Mỗi dòng có n số nguyên, cách nhau 1 khoảng trắng.

**Đầu ra**

* In các cung của đồ thị vừa nhập ra màn hình, mỗi cung trên 1 dòng theo mẫu: u v.
* Nếu có 2 cung (u1, v1) và (u2, v2) thì cung nào có u nhỏ sẽ được in ra trước. Nếu u bằng nhau thì cung nào có v nhỏ sẽ được in ra trước.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, đọc ma trận kề và lưu trực tiếp vào G.A
* Duyệt qua các ô của ma trận kề G.A. Con số trong ô G.A[u][v] cho biết số cung đi từ u đến v.

### **Ex3:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào ma trận kề của một **đồ thị vô hướng** và in danh sách kề của các đỉnh ra màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp theo mô tả ma trận kề của đồ thị. Mỗi dòng có n số nguyên, cách nhau 1 khoảng trắng.

**Đầu ra**

* In ra n dòng, mỗi dòng tương ứng với một danh sách kề.
* Dòng thứ i, in danh sách các đỉnh kề của đỉnh i. Liệt kê các đỉnh kề (có lặp lại) của i theo thứ tự tăng dần, cách nhau 1 khoảng trắng, kết thúc danh sách in thêm số 0, cách phần tử cuối cùng 1 khoảng trắng. Nếu danh sách kề của đỉnh i rỗng, chỉ in số 0 mà thôi.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, đọc ma trận kề và lưu trực tiếp vào G.A
* Duyệt qua các ô của ma trận kề G.A. Con số trong ô G.A[u][v] cho biết số cung đi từ u đến v.
* Xem mục 1.3.21 tài liệu hướng dẫn thực hành

### **Ex4:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào ma trận kề của một **đồ thị có hướng** và in danh sách kề của các đỉnh ra màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp theo mô tả ma trận kề của đồ thị. Mỗi dòng có n số nguyên, cách nhau 1 khoảng trắng.

**Đầu ra**

* In ra n dòng, mỗi dòng tương ứng với danh sách kề của 1 đỉnh.
* Dòng thứ i, in danh sách các đỉnh kề của đỉnh i. Liệt kê các đỉnh kề (có lặp lại) của i theo thứ tự tăng dần, cách nhau 1 khoảng trắng, kết thúc danh sách in thêm số 0, cách phần tử cuối cùng 1 khoảng trắng. Nếu danh sách kề của đỉnh i rỗng, chỉ in số 0 mà thôi.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, đọc ma trận kề và lưu trực tiếp vào G.A
* Duyệt qua các ô của ma trận kề G.A. Con số trong ô G.A[u][v] cho biết số cung đi từ u đến v.
* Xem thêm mục 1.3.21 trong tài liệu thực hành

### **Ex5:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào danh sách kề của các đỉnh trong một **đồ thị vô hướng** và in ma trận kề của nó màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp, mỗi dòng mô tả một danh sách kề. Dòng i chứa danh sách các đỉnh kề của đỉnh i. Các phần tử cách nhau một khoảng trắng. Mỗi dòng đều kết thúc bằng số 0.

**Đầu ra**

* In ra ma trận kề của đồ thị gồm n dòng, mỗi dòng chứa n phần tử, cách nhau 1 một khoảng trắng.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, khởi tạo đồ thị
* Đọc n danh sách kề. Với danh sách kề của đỉnh u, phần tử v của nó cho biết ta đang có cung (u, v). Chú ý, đối với đồ thị vô hướng, cung (u, v) được lưu 2 hai chỗ: v được lưu trong danh sách đỉnh kề của u và đỉnh u được lưu trong đỉnh kề của v.
* Khi đọc danh sách, sử dụng vòng lặp while (1), đọc cho tới khi gặp số 0 thì thoát vòng lặp (break).

### **Ex6:**

Viết chương trình bằng ngôn ngữ C cho phép người nhập vào danh sách kề của các đỉnh trong một **đồ thị có hướng** và in ma trận kề của nó màn hình.

**Đầu vào**

Dữ liệu đầu vào được đọc từ dòng nhập chuẩn (stdin, bàn phím) theo định dạng:

* Dòng đầu tiên chứa 1 số nguyên n mô tả số đỉnh của đồ thị
* n dòng tiếp, mỗi dòng mô tả một danh sách kề. Dòng i chứa danh sách các đỉnh kề của đỉnh i. Các phần tử cách nhau một khoảng trắng. Mỗi dòng đều kết thúc bằng số 0.

**Đầu ra**

* In ra ma trận kề của đồ thị gồm n dòng, mỗi dòng chứa n phần tử, cách nhau 1 một khoảng trắng.

**Chú ý**

* Giả sử dữ liệu đầu vào luôn hợp lệ, không cần phải kiểm tra
* Nộp toàn bộ chương trình
* Xem thêm định dạng đầu vào và đầu ra trong phần **For example**
* Ấn "Precheck" (nếu có) để kiểm tra chương trình trên các ví dụ (sai KHÔNG bị trừ điểm)
* Ấn "Check" (nếu có) để kiểm tra chương trình trên toàn bộ dữ liệu kiểm tra (sai bị TRỪ ĐIỂM)

**Gợi ý**

* Đọc n, khởi tạo đồ thị
* Đọc n danh sách kề. Với danh sách kề của đỉnh u, phần tử v của nó cho biết ta đang có cung (u, v).
* Khi đọc danh sách, sử dụng vòng lặp while (1), đọc cho tới khi gặp số 0 thì thoát vòng lặp (break).