

LAB 2

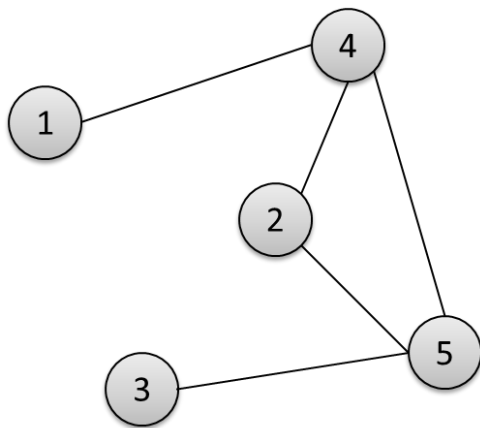
DUYỆT VÀ TÌM CÁC THÀNH PHẦN LIÊN THÔNG.

(Depth First Search)

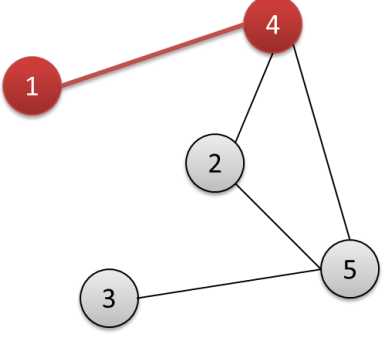
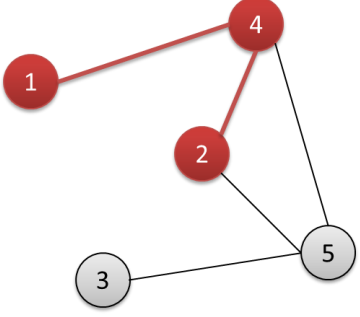
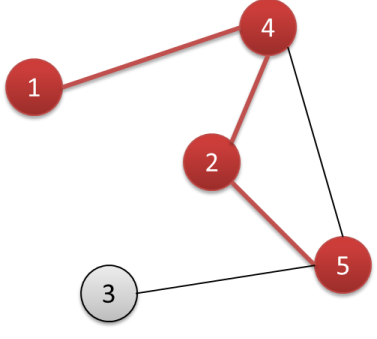
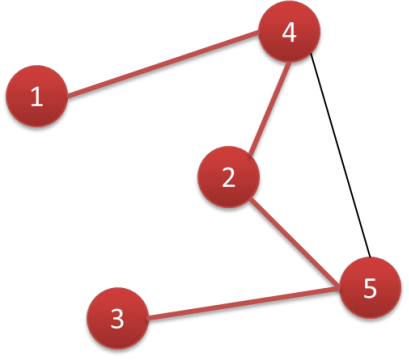
I. Đồ thị liên thông:

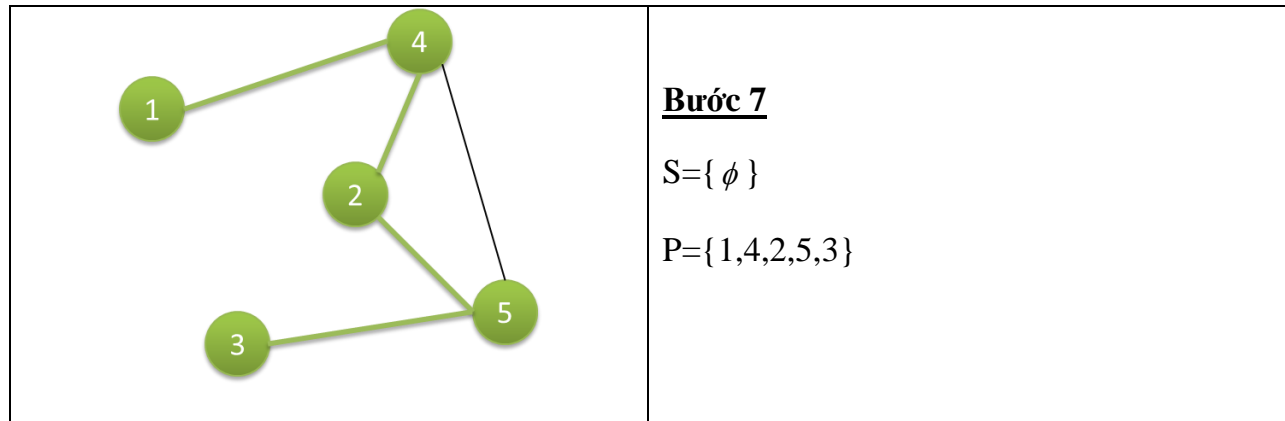
Đồ thị liên thông là đồ thị chỉ có một thành phần liên thông.

1. Thuật toán DFS



	<p><u>Bước 1</u> Khởi đầu</p> <p>$L = \{0, 0, 0, 0, 0\}$</p> <p>$S = \{1\}$</p> <p>$P = \{\phi\}$</p>
--	---

	<p><u>Bước 2</u></p> <p>$S=\{4\}$</p> <p>$P=\{1\}$</p>
	<p><u>Bước 3</u></p> <p>$S=\{2\}$</p> <p>$P=\{1,4\}$</p>
	<p><u>Bước 5</u></p> <p>$S=\{5\}$</p> <p>$P=\{1,4,2\}$</p>
	<p><u>Bước 6</u></p> <p>$S=\{3\}$</p> <p>$P=\{1,4,2,5\}$</p>



2.1 Cài đặt:

❖ Đệ Quy

Thủ tục Visit(đỉnh i , nhãn $label$)

Gán nhãn $label$ cho đỉnh i

Với mọi đỉnh j mà có cạnh nối với i

Nếu j có nhãn 0 thì

Gọi đệ quy Visit(j , $label$)

Cuối với mọi.

Cuối thủ tục.

Thủ tục	Đối tượng
<pre>void visit(int **a, int label, int n, int b[], int u) { //... } void DFS(int **a, int n) { int b[MAX]; //mang chua nhan cua dinh //nhan dinh tuong ung la 0 for(int i=0; i<n; i++) b[i]=0;</pre>	<pre>class CConnectComponents { private: int NComponents; int label[MAX]; public: CConnectComponents(void); ~CConnectComponents(void); void DFS(GRAPH g); void visit(int u, int label, GRAPH g); };</pre>

<pre> int tp=1; //... } </pre>	
----------------------------------	--

❖ **Cài đặt không đệ quy:**

<Thăm S, đánh dấu S đã thăm>;

<Đẩy S vào ngăn xếp>; // Dây chuyền đệ quy ban đầu chỉ có một đỉnh S

do {

<Lấy u khỏi ngăn xếp>; // Đang đứng ở đỉnh u

if <u có đỉnh kề chưa thăm>

{

<Chỉ chọn lấy 1 đỉnh v, là đỉnh đầu tiên kề u mà chưa được thăm>;

<Thông báo thăm v>;

<Đẩy u trở lại ngăn xếp>; // Giữ lại địa chỉ quay lui

<Đẩy tiếp v vào ngăn xếp>; // Dây chuyền duyệt sâu được "nội" thêm v

nữa

}

// Còn nếu u không có đỉnh kề chưa thăm thì ngăn xếp sẽ ngăn lại, tương ứng với quá trình lùi về của dây chuyền DFS

} **while** <Ngăn xếp khác rỗng>;

II. Bài Tập

Cho đồ thị $G = (V, U)$ tìm các thành phần liên thông của đồ thị G, hãy sử dụng lại phần hướng dẫn thực hành tuần 1.

- Viết chương trình xuất ra file kiểm tra đồ thị có liên thông hay không? Nếu có xuất ra '**LIENTHONG**' ngược lại '**KHONGLIENTHONG**', trường hợp đồ thị có thành

phần liên thông thì xuất ra có bao nhiêu thành phần liên thông và các đỉnh tương ứng với từng thành phần liên thông đó theo thứ tự thành phần liên thông nào có ít đỉnh nhất in ra trước, nếu hai thành phần liên thông có cùng số đỉnh, thì thành phần liên thông nào có đỉnh nhỏ hơn sẽ được in ra trước. Thứ tự các đỉnh trong một thành phần liên thông cũng được sắp xếp theo thứ tự tăng dần (Xem ví dụ bên dưới)

- Dùng thuật toán duyệt theo chiều sâu DFS (Depth First Search).
 - Đệ quy.
 - Không đệ quy.

Ví dụ:

input.txt	Đồ thị	output.txt
<pre> 7 0 1 0 0 0 0 1 1 0 1 0 0 0 0 0 1 0 0 0 0 1 0 0 0 0 1 0 0 0 0 0 1 0 1 0 0 0 0 0 1 0 0 1 0 1 0 0 0 0 </pre>	<pre> graph TD 0 --- 6 0 --- 1 6 --- 5 1 --- 2 5 --- 4 4 --- 3 2 --- 3 </pre>	<pre> KHONGLIENTHONG 2 3 4 5 0 1 2 6 ... </pre>

Tài liệu tham khảo :

- [1.] Huỳnh Lê Tấn Tài, Bài tập thực hành Lý thuyết đồ thị, Khoa CNTT-TUD, ĐH Tôn Đức Thắng.
- [2.] Bài tập thực hành Lý thuyết đồ thị, Khoa CNTT, ĐH Khoa Học Tự Nhiên.