Chương 2. TÍNH LIÊN THÔNG CỦA ĐỒ THỊ

ThS. Nguyễn Chí Hiếu

2019

NỘI DUNG

Các khái niệm cơ bản

Định nghĩa 1

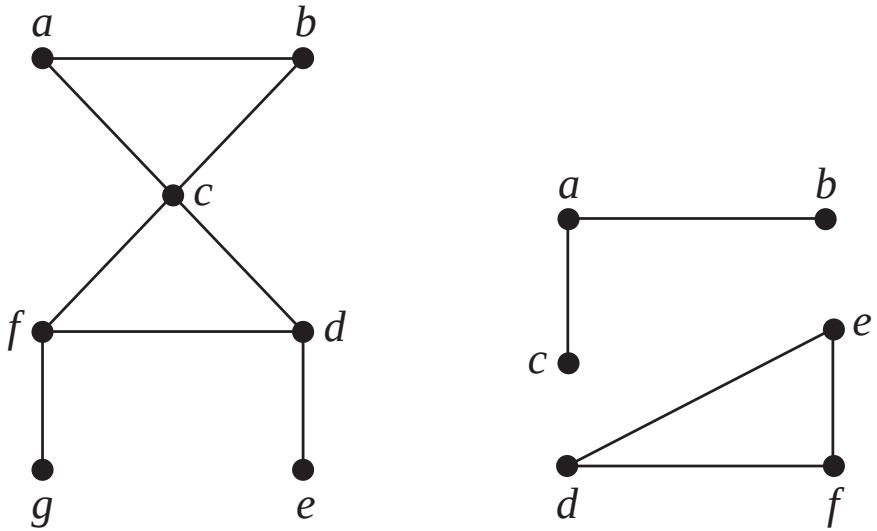
Cho đồ thị $G = (V, E)$. Trên tập hợp V, ta định nghĩa một quan hệ $\sim$ như sau: $\forall u, v \in V, u \sim v \Leftrightarrow u = v$ hay có một đường đi từ u đến v.

- Quan hệ $\sim$ chia V thành các lớp tương đương. Mỗi lớp tương đương được gọi là một **thành phần liên thông** *(connected component)* của G.
- Nếu G chỉ có **một** thành phần liên thông thì G được gọi là **đồ thị liên thông** *(connected graph)*.

Các khái niệm cơ bản

Ví dụ 1

Cho G_1 là đồ thị liên thông và G_2 là đồ thị không liên thông.



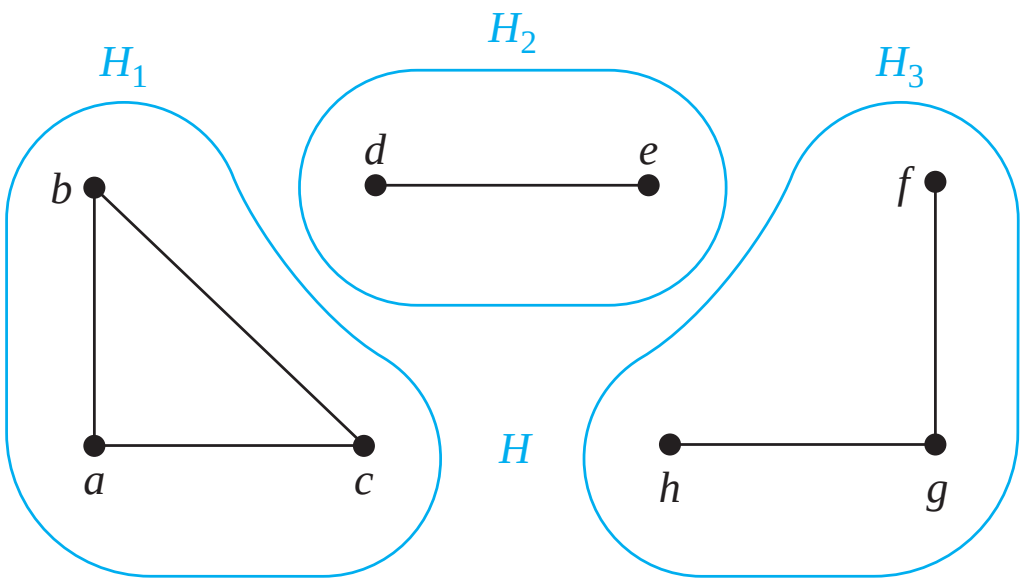
G_1 G_2

Hình 1: Đồ thị G_1 và G_2.

Các khái niệm cơ bản

Ví dụ 2

Cho H là đồ thị không liên thông và có 3 thành phần liên thông.



Hình 2: Đồ thị H và 3 thành phần liên thông H_1, H_2 và H_3.

Các khái niệm cơ bản

Định nghĩa 2

Đồ thị có hướng là **liên thông mạnh** *(strongly connected)* nếu có một đường đi từ u đến v và từ v đến u với mọi u, v là đỉnh thuộc đồ thị.

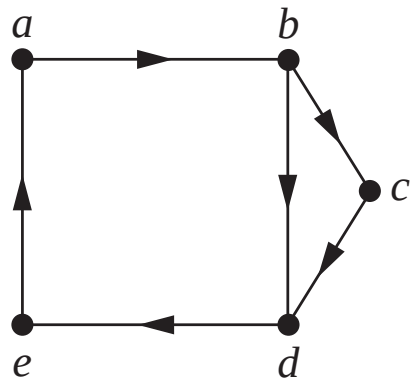
Định nghĩa 3

Đồ thị có hướng là **liên thông yếu** *(weakly connected)* nếu có một đường đi giữa hai đỉnh bất kỳ trong đồ thị vô hướng tương ứng với đồ thị đã cho.

Các khái niệm cơ bản

Ví dụ 3

Cho G_1 là đồ thị liên thông mạnh.

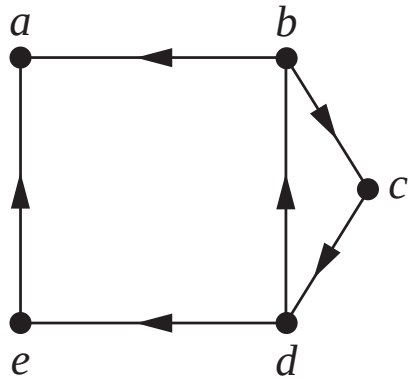


Hình 3: Đồ thị G_1.

Các khái niệm cơ bản

Ví dụ 4

Cho G_2 là đồ thị liên thông yếu.



Hình 4: Đồ thị G_2.

Nhận xét

- Không có bất kỳ đường đi nào từ đỉnh a đến các đỉnh khác trong đồ thị

Các thuật toán tìm kiếm/duyệt đồ thị

Ý tưởng

Các thuật toán tìm kiếm/duyệt đồ thị bắt đầu tại một đỉnh i bất kỳ của đồ thị. Sau đó, chọn đỉnh j là đỉnh **kề** với đỉnh i và lặp lại quá trình thực hiện đối với j.

Dựa vào thứ tự duyệt đỉnh thuật toán được chia thành hai loại:

- Thuật toán tìm kiếm theo chiều sâu *(Depth First Search - DFS)*.
- Thuật toán tìm kiếm theo chiều rộng *(Breadth First Search - BFS)*.

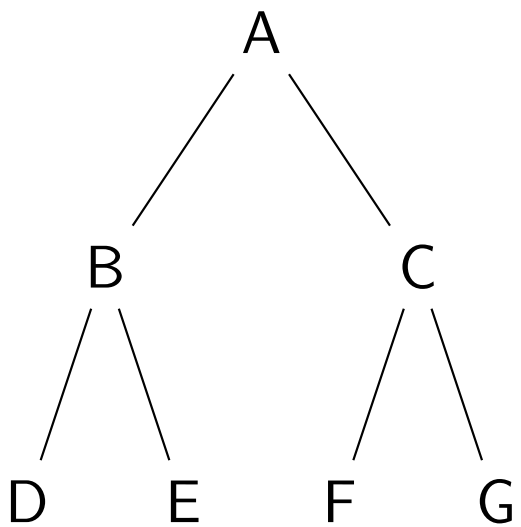
Thuật toán tìm kiếm theo chiều sâu (DFS)

Ý tưởng

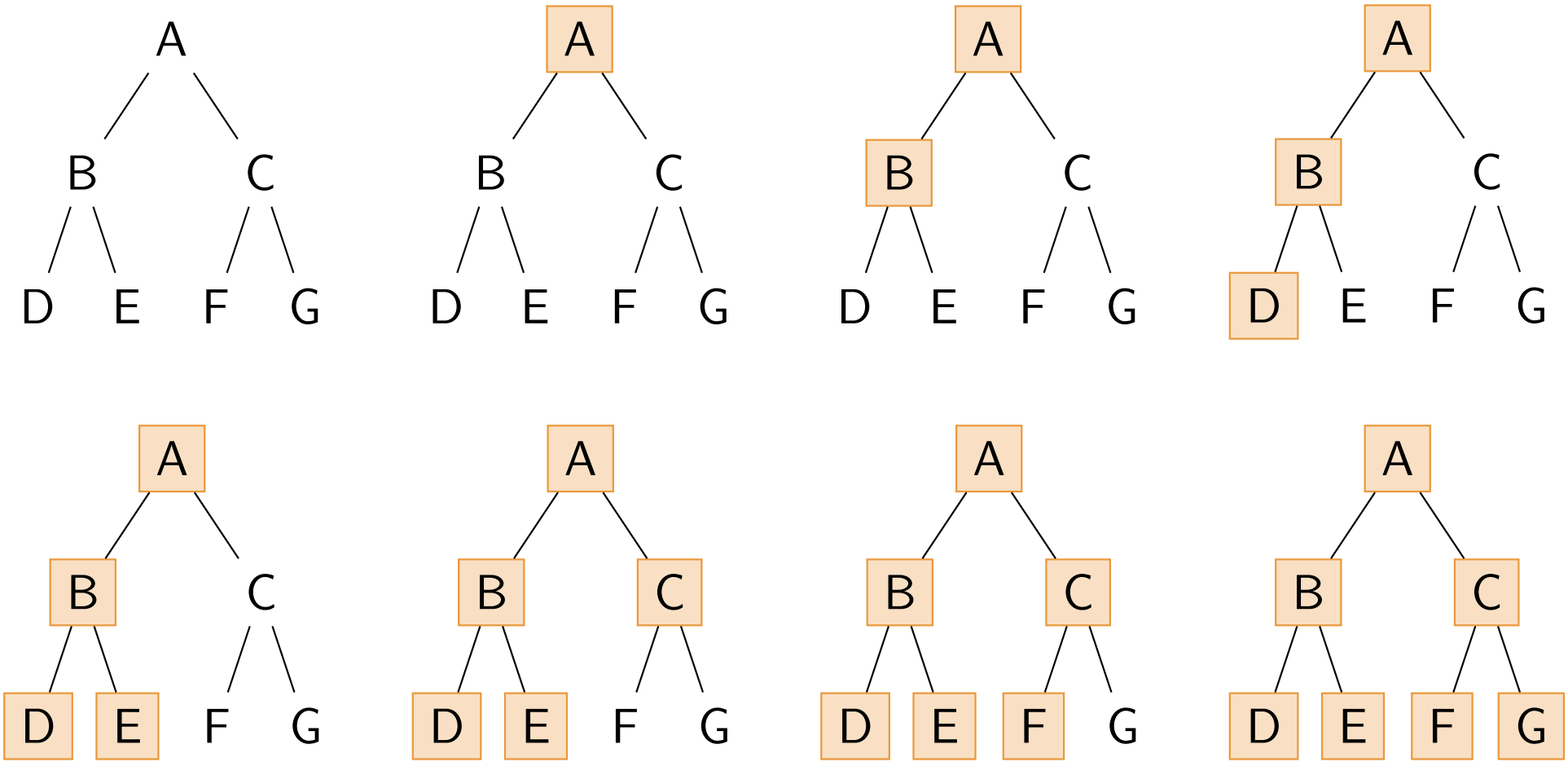
Tại mỗi đỉnh đang xét, thuật toán sẽ mở rộng các đỉnh chưa xét theo mức sâu nhất.

Ví dụ 5

Áp dụng thuật toán DFS duyệt tất cả đỉnh của đồ thị.



Thuật toán tìm kiếm theo chiều sâu (DFS)



Thuật toán tìm kiếm theo chiều sâu

Thuật toán 1: **DFS(i)**

- Đầu vào: đỉnh i đang xét.
- Đầu ra: đường đi từ i qua tất cả các đỉnh theo chiều sâu.

```
1  visited[i] ← true
2
3  foreach vertex j ∈ V
4     if edge(i, j) ∈ E and visited[j] = false
5        DFS(j)
```

Giải thích

- Dòng 1: đánh dấu đỉnh *i* đã được duyệt.
- Dòng 3 $\rightarrow$ 4: tìm đỉnh *j* kề với đỉnh *i* và chưa được duyệt.
- Dòng 5: gọi đệ quy thuật toán DFS tại đỉnh *j*.

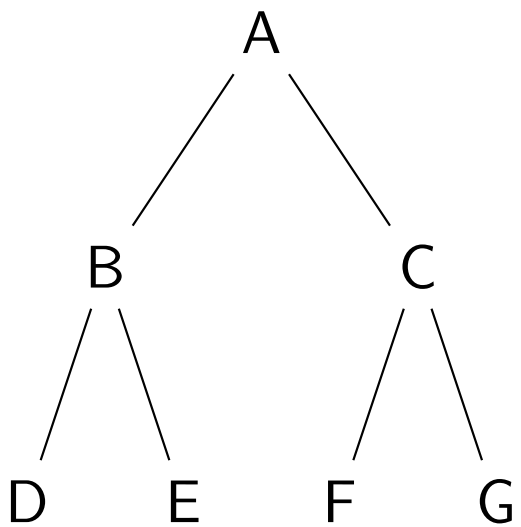
Thuật toán tìm kiếm theo chiều rộng (BFS)

Ý tưởng

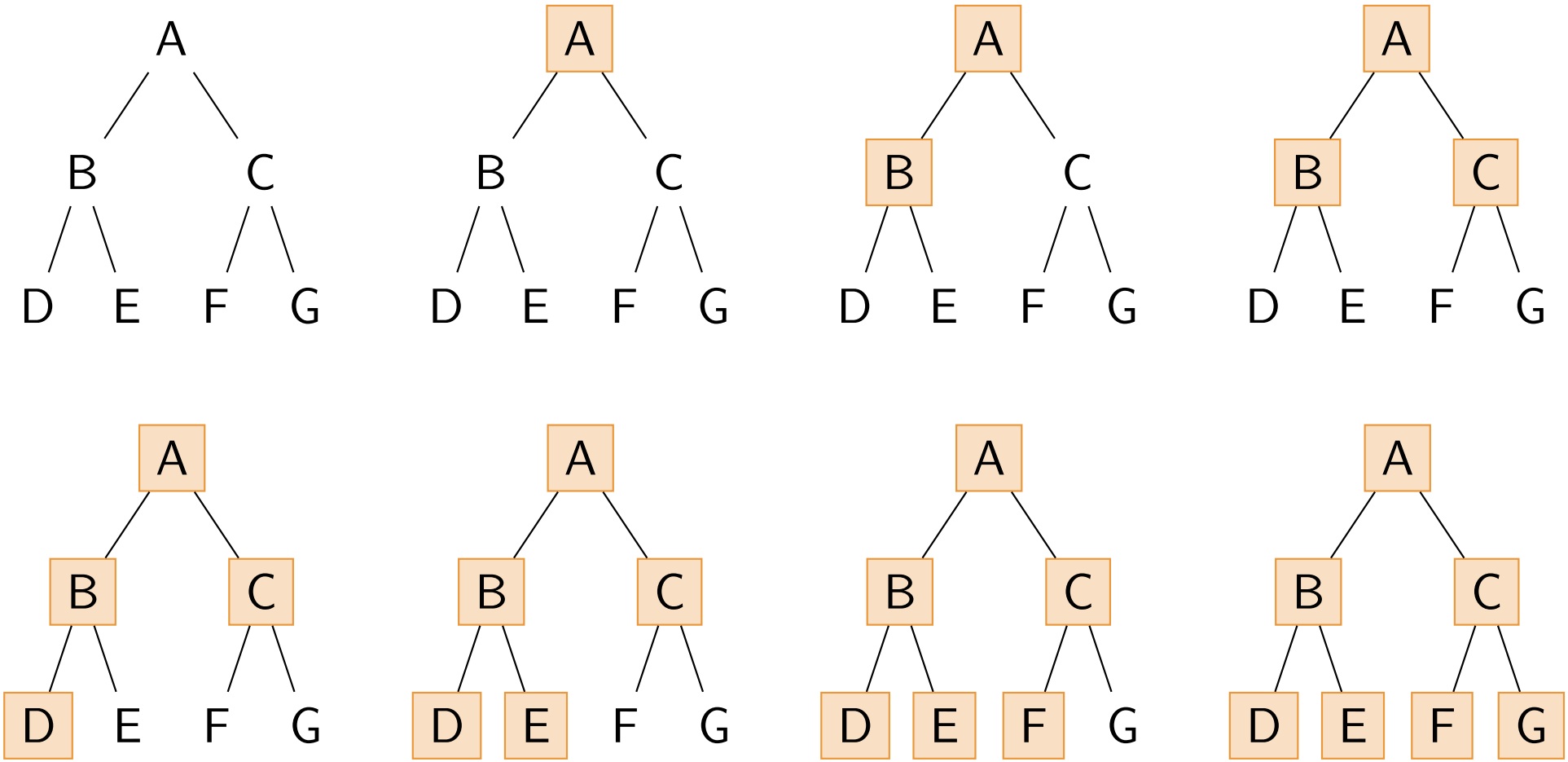
Tại mỗi đỉnh đang xét, thuật toán sẽ mở rộng các đỉnh chưa xét theo mức sâu nhất.

Ví dụ 6

Áp dụng thuật toán BFS duyệt tất cả đỉnh của đồ thị.



Thuật toán tìm kiếm theo chiều rộng (BFS)



Thuật toán kiểm tra đồ thị liên thông

Thuật toán 2: **IsConnected()**

- Đầu vào: đồ thị
- Đầu ra: số thành phần liên thông

```
1  components ← 0
2  foreach vertice i ∈ V
3     visited[i] ← -1
4
5  foreach vertice i ∈ V
6     if visited[i] = -1
7        components ← components + 1
8        Visit(i, components)
9
10 return components = 1
```

Thuật toán kiểm tra đồ thị liên thông

Giải thích

- Dòng 1: khởi tạo nhãn thành phần liên thông bằng 0.
- Dòng 2: khởi tạo mảng lưu trữ trạng thái các đỉnh.
- Dòng 5 $\rightarrow$ 8 : duyệt qua tất cả đỉnh chưa duyệt.
- Dòng 7: tăng nhãn thành phần liên thông.
- Dòng 8: gọi hàm đệ quy `Visit()` áp dụng thuật toán tìm kiếm DFS hay BFS.
- Dòng 10: nếu nhãn thành phần liên thông là 1, trả về `true` *(đồ thị liên thông)*.

Thuật toán kiểm tra đồ thị liên thông

Thuật toán 3: **Visit(i, components)**

- Đầu vào: đỉnh i đang xét và nhãn thành phần liên thông
- Đầu ra: đường đi từ i qua tất cả các đỉnh theo chiều sâu.

```
1  visited[i] ← components
2
3  foreach vertex j ∈ V
4     if edge(i, j) ∈ E and visited[j] = -1
5        Visit(j, components)
```

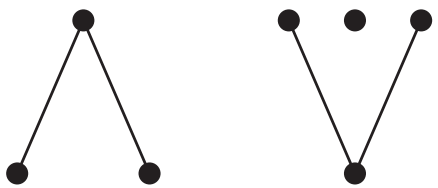
Ứng dụng của thuật toán tìm thành phần liên thông

- Kiểm tra đồ thị liên thông mạnh.
- Kiểm tra đồ thị liên thông trước khi thực hiện một số thuật toán tìm đường đi hay chu trình trong đồ thị.

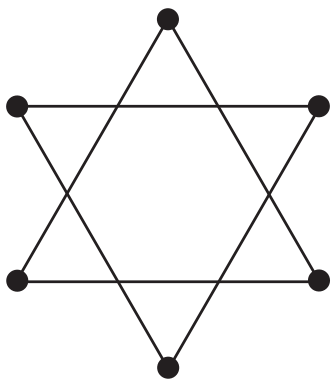
Bài tập

1. Cho đồ thị G_1 và G_2, xác định các đồ thị có liên thông hay không? Nếu đồ thị không liên thông, xác định số thành phần liên thông của đồ thị.

a)



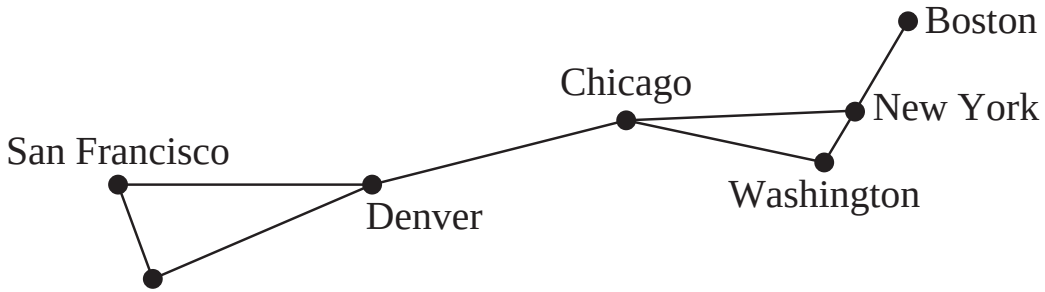
b)



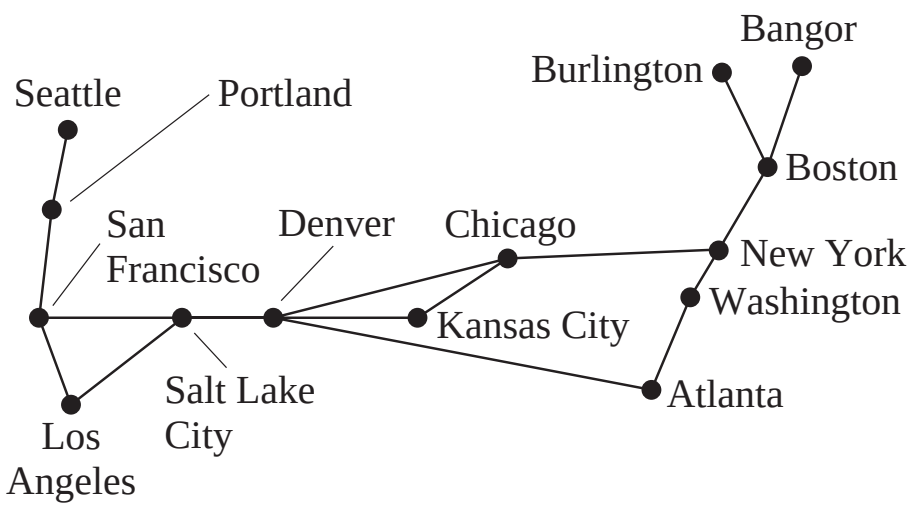
Bài tập

2. Mạng máy tính giữa các thành phố cần được cung cấp đường truyền dự phòng để không bị ảnh hưởng khi có sự cố. Đối với các mạng máy tính sau đây, xác định những đường truyền nào cần phải dự phòng.

a)



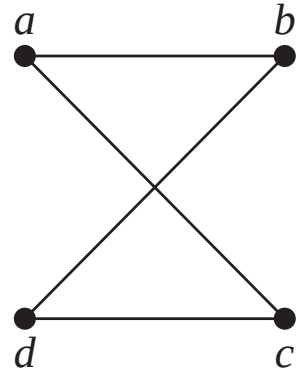
b)



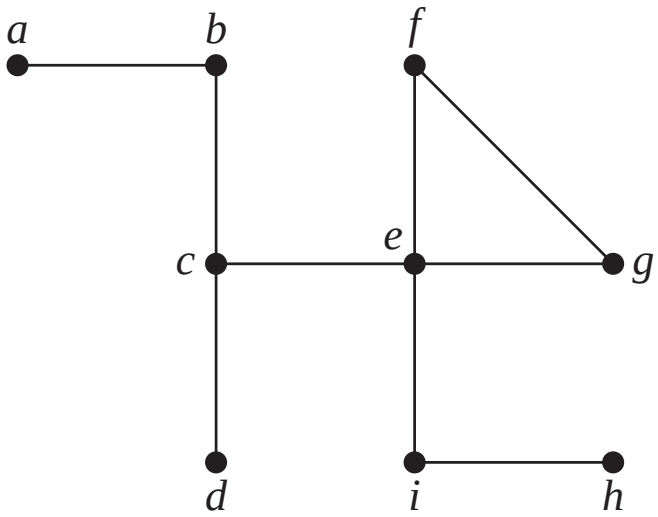
Bài tập

3. Cho đồ thị G_1 và G_2, in thứ tự duyệt các đỉnh theo thuật toán DFS và BFS. Thuật toán bắt đầu tại đỉnh c.

a)



b)



Tài liệu tham khảo


Adrian Bondy, U.S.R. Murty, *Graph Theory*, Springer, 2008.


Kenneth H. Rosen, *Discrete Mathematics and its Applications, 7th Edidion*, McGraw-Hill, 2011.


Nguyễn Cam, Chu Đức Khánh, *Lý thuyết đồ thị*, NXB Đại học Quốc gia Tp Hồ Chí Minh, 2008.


Nguyễn Đức Nghĩa, Nguyễn Tô Thành, *Toán rời rạc*, NXB Đại học Quốc gia Hà Nội, 2003.


Reinhard Diestel, *Graph Theory*, Springer, 2005.