

# Cấu trúc Đồ thị

## Phần 1: Các khái niệm cơ bản

Nguyễn Thanh Bình

Viện ĐTVT – ĐHBK Hà Nội

2020

# Nội dung chính

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

1 Cơ bản về đồ thị

2 Cài đặt đồ thị

3 Phép duyệt đồ thị

# Các khái niệm cơ bản

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Đồ thị (Graph)

Đồ thị  $G$  là một cấu trúc gồm hai thành phần:

- $V = \{v_0, v_1, \dots, v_{m-1}\}$ : tập hữu hạn gồm  $m$  đỉnh (nút, hay điểm).
- $E = \{e_0, e_1, \dots, e_{n-1}\}$ , với  $e_i = (v_j, v_k)$ : tập hữu hạn gồm  $n$  cạnh (hay cung) nối các cặp đỉnh.

Kí hiệu đầy đủ cho đồ thị  $G$  là  $G(V, E)$ .

# Các khái niệm cơ bản

Cấu trúc Đồ thị

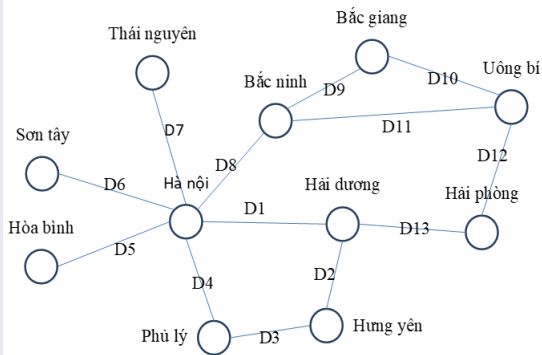
Nguyễn  
Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

Ví dụ: Bản đồ giao thông là một đồ thị, trong đó các thành phố biểu diễn các đỉnh, còn các đường giao thông nối giữa chúng biểu diễn các cạnh.



# Các khái niệm cơ bản

Cấu trúc Đồ thị

Nguyễn  
Thanh Bình

Cơ bản về đồ thị

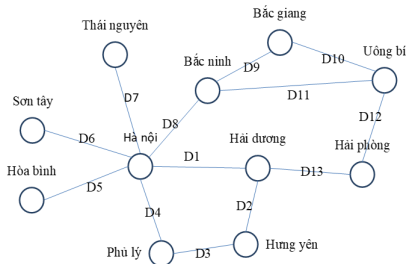
Cài đặt đồ thị

Phép duyệt đồ thị

## Đỉnh kề và cạnh kề

Giả sử  $v_i$  là một đỉnh của đồ thị  $G(V,E)$ . Khi đó, ta gọi  $v_j$  là *đỉnh kề* của  $v_i$  nếu  $(v_i, v_j) \in E$ .

Đồng thời, bản thân  $(v_i, v_j)$  cũng là *cạnh kề* của  $v_i$  và  $v_j$ .



# Các khái niệm cơ bản

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Đường đi (path)

Đường đi giữa hai đỉnh  $v_b, v_e$ , ký hiệu  $P(v_b, v_e)$ , của đồ thị  $G(V, E)$  là tập các cạnh liên tiếp nhau nối liền hai đỉnh đó. Tức là, tồn tại tập đỉnh  $v_1 = v_b, v_2, \dots, v_k = v_e$ , sao cho  $\forall i = 1..(k-1), (v_i, v_{i+1}) \in E$ . Ta có:

$$P(v_b, v_e) = [(v_1, v_2), (v_2, v_3), \dots, (v_{k-1}, v_k)].$$

$v_b$  và  $v_e$  lần lượt được gọi là *đỉnh đầu* và *đỉnh cuối* của đường đi  $P$ . Để cho ngắn gọn, đường đi sẽ được biểu diễn bằng một dãy các đỉnh như sau:

$$P(v_b, v_e) = (v_1, v_2, v_3, \dots, v_{k-1}, v_k).$$

# Các khái niệm cơ bản

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

## Ví dụ về Đường đi

Ở đồ thị hình bên, để nối hai thành phố Sơn tây và Uông bí ta có thể có một số đường đi như:

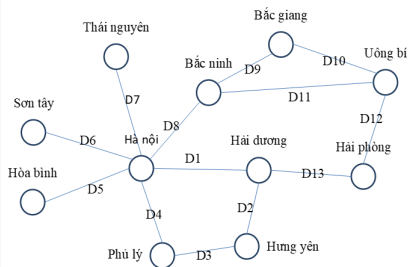
$P_1 = (D6, D8, D11),$

$P_2 = (D6, D1, D13, D12).$

Hoặc nếu biểu diễn dưới dạng dãy các đỉnh như sau:

$P_1 = (\text{Sơn tây}, \text{Hà nội}, \text{Bắc ninh}, \text{Uông bí}),$

$P_2 = (\text{Sơn tây}, \text{Hà nội}, \text{Hải dương}, \text{Hải phòng}, \text{Uông bí}).$



# Các khái niệm cơ bản

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

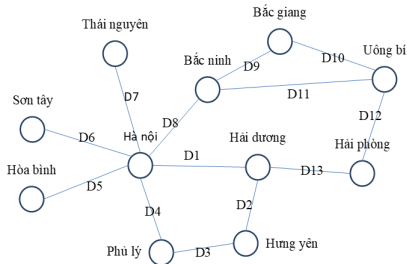
Phép duyệt đồ thị

## Chu trình (circuit)

Chu trình là một đường đi khép kín (tức là đỉnh đầu và đỉnh cuối của nó trùng nhau) và không có cạnh nào xuất hiện quá một lần.

## Ví dụ

Trong đồ thị hình bên có một số chu trình như:  
C1= (Hà nội, Phủ lý, Hưng yên, Hải dương, Hà nội),  
C2 = (Bắc giang, Uông bí, Bắc ninh, Bắc giang).





# Các loại đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Đồ thị vô hướng (undirected graph)

Là đồ thị mà chiều của mỗi cạnh không quan trọng, mỗi cạnh là một tập hợp của hai đỉnh (tức là cạnh  $(u, v)$  trùng với cạnh  $(v, u)$ ).

Ví dụ như đồ thị sơ đồ giao thông ở trên là đồ thị vô hướng.

## Đồ thị có hướng (directed graph)

Là đồ thị mà mỗi cạnh đều có một chiều nhất định, hay nói cách khác mỗi cạnh là một tập có thứ tự của hai đỉnh (tức là cạnh  $(u, v)$  khác cạnh  $(v, u)$ ). Khi đó, mỗi cạnh còn được gọi là một *cung* (arc).

# Các loại đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Đồ thị có trọng số (weighted graph)

Là đồ thị mà mỗi cạnh mang một giá trị nào đó mà được gọi là *trọng số* của cạnh đó. Còn ngược lại, nếu các cạnh của đồ thị không mang trọng số nào thì đồ thị đó được gọi là *không có trọng số*.

# Các loại đồ thị

Cấu trúc Đồ  
thị

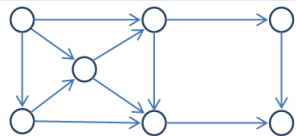
Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

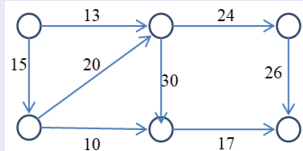
Cài đặt đồ thị

Phép duyệt đồ  
thị

Đồ thị có hướng



Đồ thị có trọng số



# Một số thao tác cơ bản

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

- **Duyệt:** tương tự như phép duyệt cây, phép duyệt đồ thị cũng muốn truy nhập tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần. Có hai phương pháp duyệt đồ thị mà sẽ được trình bày sau này là *duyet theo chiều sâu* và *duyet theo chiều rộng*.

# Một số thao tác cơ bản

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

- **Duyệt:** tương tự như phép duyệt cây, phép duyệt đồ thị cũng muốn truy nhập tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần. Có hai phương pháp duyệt đồ thị mà sẽ được trình bày sau này là *duyet theo chiều sâu* và *duyet theo chiều rộng*.
- **Tìm kiếm một đỉnh hoặc một cạnh:** là thao tác muốn tìm một đỉnh hay một cạnh thỏa mãn một điều kiện nào đó, ví như tìm đỉnh có giá trị cho trước, hay tìm cạnh có trọng số cho trước. Các giải thuật tìm kiếm này thường dựa vào các phương pháp duyệt cây.

# Một số thao tác cơ bản

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

- **Duyệt:** tương tự như phép duyệt cây, phép duyệt đồ thị cũng muốn truy nhập tất cả các đỉnh của đồ thị, mỗi đỉnh đúng một lần. Có hai phương pháp duyệt đồ thị mà sẽ được trình bày sau này là *duyet theo chiều sâu* và *duyet theo chiều rộng*.
- **Tìm kiếm một đỉnh hoặc một cạnh:** là thao tác muốn tìm một đỉnh hay một cạnh thỏa mãn một điều kiện nào đó, ví như tìm đỉnh có giá trị cho trước, hay tìm cạnh có trọng số cho trước. Các giải thuật tìm kiếm này thường dựa vào các phương pháp duyệt cây.
- **Tìm đường đi ngắn nhất:** bài toán này thực ra có thể lại bao gồm một số bài toán con như:
  - Tìm đường đi ngắn nhất giữa hai điểm.
  - Tìm đường đi ngắn nhất giữa một điểm và tất cả các điểm còn lại.

# Cài đặt đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

Theo cấu trúc lưu trữ, có hai phương pháp cài đặt chính cho cấu trúc đồ thị:

- *Sử dụng cấu trúc lưu trữ tuần tự*: phương pháp này sử dụng cấu trúc mảng để lưu trữ các đỉnh và các cạnh của đồ thị.

# Cài đặt đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

Theo cấu trúc lưu trữ, có hai phương pháp cài đặt chính cho cấu trúc đồ thị:

- *Sử dụng cấu trúc lưu trữ tuần tự*: phương pháp này sử dụng cấu trúc mảng để lưu trữ các đỉnh và các cạnh của đồ thị.
- *Sử dụng cấu trúc lưu trữ móc nối*: phương pháp này dùng cấu trúc lưu trữ móc nối mà gồm các nút để biểu diễn các đỉnh, và các mối nối để biểu diễn các cạnh/cung của đồ thị.



# So sánh hai phương pháp cài đặt

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

- Sử dụng cấu trúc lưu trữ tuần tự:
  - Ưu điểm: Tổ chức lưu trữ khá đơn giản. Các thao tác cài đặt có tốc độ khá nhanh do khả năng truy nhập trực tiếp của cấu trúc mảng.
  - Nhược điểm: Thường chỉ thích hợp khi kích thước đồ thị nhỏ (trong khoảng vài chục đỉnh và cạnh). Gây ra sự lãng phí kích thước lưu trữ lớn khi kích thước đồ thị lớn.

# So sánh hai phương pháp cài đặt

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

- Sử dụng cấu trúc lưu trữ tuần tự:
  - Ưu điểm: Tổ chức lưu trữ khá đơn giản. Các thao tác cài đặt có tốc độ khá nhanh do khả năng truy nhập trực tiếp của cấu trúc mảng.
  - Nhược điểm: Thường chỉ thích hợp khi kích thước đồ thị nhỏ (trong khoảng vài chục đỉnh và cạnh). Gây ra sự lãng phí kích thước lưu trữ lớn khi kích thước đồ thị lớn.
- Sử dụng cấu trúc lưu trữ móc nối:
  - Ưu điểm: Giải quyết các nhược điểm của cách sử dụng cấu trúc tuần tự.
  - Nhược điểm: Các thao tác thường có độ phức tạp cao, do cơ chế truy nhập tuần tự của cấu trúc móc nối. Để giảm thiểu hạn chế này, việc tăng cường các điểm truy nhập đôi khi là cần thiết, nhưng cũng cần phải cân bằng với mức độ phức tạp và lãng phí bộ nhớ mà biện pháp này gây ra.

# Sử dụng cấu trúc lưu trữ tuần tự

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

Giả sử cần cài đặt đồ thị  $G(V, E)$ , với  $V = (v_0, v_1, \dots, v_{M-1})$ , và  $E = (e_0, e_1, \dots, e_{N-1})$ . Để lưu trữ các đỉnh, ta sử dụng một mảng một chiều  $AV[M]$ . Còn để lưu trữ các cạnh, lại có thể có một trong hai cách:

- *Sử dụng ma trận đỉnh kề*: trong cách này, sử dụng mảng hai chiều  $AE[M \times M]$  để lưu trữ các cạnh.

# Sử dụng cấu trúc lưu trữ tuần tự

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

Giả sử cần cài đặt đồ thị  $G(V, E)$ , với  $V = (v_0, v_1, \dots, v_{M-1})$ , và  $E = (e_0, e_1, \dots, e_{N-1})$ . Để lưu trữ các đỉnh, ta sử dụng một mảng một chiều  $AV[M]$ . Còn để lưu trữ các cạnh, lại có thể có một trong hai cách:

- *Sử dụng ma trận đỉnh kề*: trong cách này, sử dụng mảng hai chiều  $AE[M \times M]$  để lưu trữ các cạnh.
- *Sử dụng ma trận cạnh kề*: Trong cách này lại cần có thêm hai cấu trúc mảng: mảng thứ nhất một chiều  $E[N]$  để lưu các cạnh, và mảng thứ hai hai chiều  $AE[M \times N]$  là mảng kiểu logic để lưu trạng thái các cạnh kề. Cụ thể là nếu đỉnh  $v_i$  kề với cạnh  $e_j$  thì  $AE[i, j] = \text{True (T)}$ . Còn trái lại thì  $AE[i, j] = \text{False}$  (bỏ trống). Trong trường hợp đặc biệt, nếu đồ thị  $G$  là đồ thị không có trọng số, tức là giá trị các cạnh là không quan trọng, thì mảng  $E$  có thể bỏ đi được.

# Sử dụng cấu trúc lưu trữ tuần tự

Cấu trúc Đồ thị

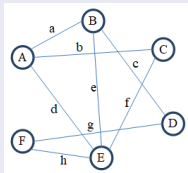
Nguyễn  
Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

Ví dụ: Giải sử với đồ thị



Ma trận đỉnh kề

	A	B	C	D	E	F
A		T	T		T	
B	T			T	T	
C	T				T	
D		T				T
E	T	T	T			T
F				T	T	

Ma trận cạnh kề

	a	b	c	d	e	f	g	h
A	T	T		T				
B	T		T		T			
C		T				T		
D			T				T	
E				T	T	T		T
F							T	T

# Sử dụng cấu trúc lưu trữ móc nối

## Ý tưởng chung:

- *Tập các đỉnh được biểu diễn bằng một danh sách móc nối:* Tổ chức móc nối cũng có nhiều cách như: móc nối đơn, móc nối kép, móc nối thẳng, móc nối vòng, v.v. trong đó, cấu trúc móc nối đơn luôn là lựa chọn đầu tiên.

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

# Sử dụng cấu trúc lưu trữ móc nối

## Ý tưởng chung:

- *Tập các đỉnh được biểu diễn bằng một danh sách móc nối:* Tổ chức móc nối cũng có nhiều cách như: móc nối đơn, móc nối kép, móc nối thẳng, móc nối vòng, v.v. trong đó, cấu trúc móc nối đơn luôn là lựa chọn đầu tiên.
- *Tập các cạnh cũng được biểu diễn bằng danh sách móc nối:* do bản thân mỗi cạnh lại cần có thông tin của cặp đỉnh kề nó và bản thân thông tin của cạnh đó (như trọng số của cạnh đó nếu đồ thị có trọng số), nên biểu diễn các cạnh này cũng lại có thể có nhiều cách khác nhau như sau:

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

# Sử dụng cấu trúc lưu trữ móc nối

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

## Ý tưởng chung:

- *Tập các đỉnh được biểu diễn bằng một danh sách móc nối:* Tổ chức móc nối cũng có nhiều cách như: móc nối đơn, móc nối kép, móc nối thẳng, móc nối vòng, v.v. trong đó, cấu trúc móc nối đơn luôn là lựa chọn đầu tiên.
- *Tập các cạnh cũng được biểu diễn bằng danh sách móc nối:* do bản thân mỗi cạnh lại cần có thông tin của cặp đỉnh kề nó và bản thân thông tin của cạnh đó (như trọng số của cạnh đó nếu đồ thị có trọng số), nên biểu diễn các cạnh này cũng lại có thể có nhiều cách khác nhau như sau:
  - *Tổ chức danh sách đỉnh kề:* trong cách tổ chức này, các đỉnh kề với một đỉnh sẽ nằm trong một danh sách móc nối.



# Sử dụng cấu trúc lưu trữ móc nối

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

## Ý tưởng chung:

- *Tập các đỉnh được biểu diễn bằng một danh sách móc nối*: Tổ chức móc nối cũng có nhiều cách như: móc nối đơn, móc nối kép, móc nối thẳng, móc nối vòng, v.v. trong đó, cấu trúc móc nối đơn luôn là lựa chọn đầu tiên.
- *Tập các cạnh cũng được biểu diễn bằng danh sách móc nối*: do bản thân mỗi cạnh lại cần có thông tin của cặp đỉnh kề nó và bản thân thông tin của cạnh đó (như trọng số của cạnh đó nếu đồ thị có trọng số), nên biểu diễn các cạnh này cũng lại có thể có nhiều cách khác nhau như sau:
  - *Tổ chức danh sách đỉnh kề*: trong cách tổ chức này, các đỉnh kề với một đỉnh sẽ nằm trong một danh sách móc nối.
  - *Tổ chức danh sách cạnh kề*: trong cách tổ chức này, các cạnh kề với một đỉnh sẽ nằm trong một danh sách móc nối.

# Sử dụng cấu trúc lưu trữ móc nối

Cấu trúc Đồ thị

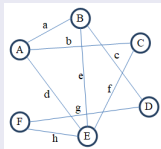
Nguyễn  
Thanh Bình

Cơ bản về đồ thị

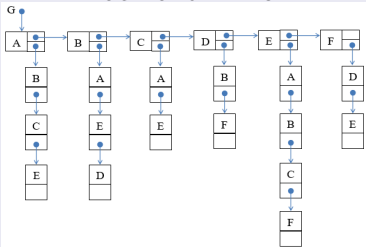
Cài đặt đồ thị

Phép duyệt đồ thị

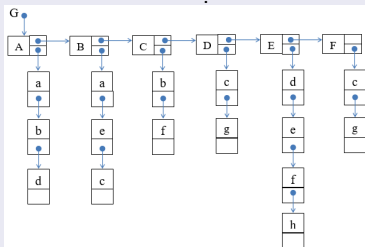
Ví dụ: Giả sử với đồ thị



Móc nối đỉnh kề



Móc nối cạnh kề



# Phép duyệt đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

# Phép duyệt đồ thị

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

## Hai phương pháp duyệt cơ bản:

- Duyệt theo chiều sâu
- Duyệt theo chiều rộng

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Ý tưởng giải thuật

Đây là một giải thuật đệ quy có các bước cơ bản như sau:

① Đặt  $v = v_s$ .

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Ý tưởng giải thuật

Đây là một giải thuật đệ quy có các bước cơ bản như sau:

- 1 Đặt  $v = v_s$ .
- 2 Thăm nút  $v$ .

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Ý tưởng giải thuật

Đây là một giải thuật đệ quy có các bước cơ bản như sau:

- 1 Đặt  $v = v_s$ .
- 2 Thăm nút  $v$ .
- 3 Với mỗi nút  $v_a$  là kề của  $v$  mà chưa được thăm, lặp lại bước 2 cho  $v_a$ .

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Ý tưởng giải thuật

Đây là một giải thuật đệ quy có các bước cơ bản như sau:

- 1 Đặt  $v = v_s$ .
- 2 Thăm nút  $v$ .
- 3 Với mỗi nút  $v_a$  là kề của  $v$  mà chưa được thăm, lặp lại bước 2 cho  $v_a$ .
- 4 (Điểm dừng): giải thuật kết thúc khi đồ thị không còn nút nào cần phải thăm nữa.



# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Thủ tục

Gọi thủ tục cài đặt cho giải thuật là *FullDepthSearch*( $G, vs$ ), với  $vs$  là đỉnh bắt đầu mà thuộc tập các đỉnh  $V$  của đồ thị  $G$ .

```
void FullDepthSearch( $G, vs$ )
```

```
{
```

```
     $v = vs$ ;
```

```
    DepthFirstSearch( $G, v$ );
```

```
}
```

# Thủ tục *DepthFirstSearch*( $G, v$ )

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

$\text{DepthFirstSearch}(G, v)$  là thủ tục đệ quy duyệt theo chiều sâu có dạng như sau:

```
void DepthFirstSearch( $G, v$ )
```

```
{  
    if ( $v == \text{NULL}$ ) return; //Điểm dừng  
    visit( $v$ );  
    for each unvisited adjacent  $va$  to  $v$   
        DepthFirstSearch( $G, va$ );  
}
```

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

**Lưu ý:** đối với giải thuật trên, có hai trường hợp đặc biệt cần lưu ý:

- *Đồ thị có tồn tại chu trình*: cần cập nhật đúng các trạng thái *đã thăm/chưa thăm* của từng nút trong quá trình duyệt để tránh rơi vào vòng lặp vô hạn.

# Phép duyệt đồ thị - Duyệt theo chiều sâu

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

**Lưu ý:** đối với giải thuật trên, có hai trường hợp đặc biệt cần lưu ý:

- *Đồ thị có tồn tại chu trình:* cần cập nhật đúng các trạng thái *đã thăm/chưa thăm* của từng nút trong quá trình duyệt để tránh rơi vào vòng lặp vô hạn.
- *Đồ thị có các nút cô lập:* các nút này có thể bị bỏ quên khi duyệt, nên trong giải thuật cần cân nhắc để đưa thêm các nút này vào.

# Duyệt theo chiều sâu - Ví dụ

Cấu trúc Đồ thị

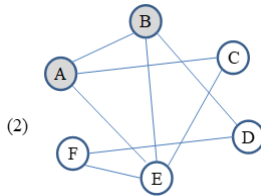
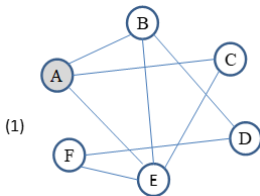
Nguyễn  
Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

Các hình dưới đây minh họa hoạt động của giải thuật trên, trong đó các nút xám là nút đã được thăm, còn các nút trắng là chưa được thăm.



# Duyệt theo chiều sâu - Ví dụ

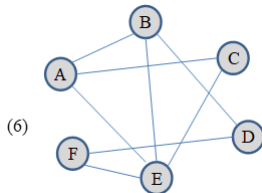
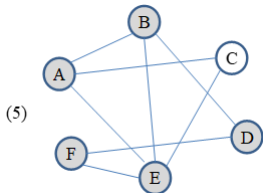
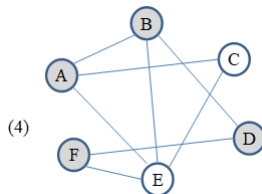
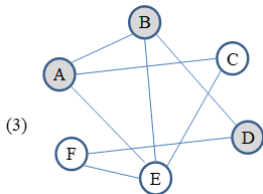
Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị



# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

## Ý tưởng của giải thuật duyệt theo chiều rộng

- 1 Thăm nút  $v_s$ .

# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

## Ý tưởng của giải thuật duyệt theo chiều rộng

- 1 Thăm nút  $v_s$ .
- 2 Tìm  $VA$  là tập các nút kề của  $v_s$  mà chưa được thăm.



# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

## Ý tưởng của giải thuật duyệt theo chiều rộng

- 1 Thăm nút  $v_s$ .
- 2 Tìm  $VA$  là tập các nút kề của  $v_s$  mà chưa được thăm.
- 3 Thăm lần lượt các nút trong  $VA$ .

# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Bài toán duyệt đồ thị

Cho trước đồ thị  $G(V, E)$  và nút xuất phát  $v_s \in V$ . Tìm cách duyệt tất cả các đỉnh của  $G$ , mỗi đỉnh đúng một lần, bắt đầu từ nút  $v_s$ .

## Ý tưởng của giải thuật duyệt theo chiều rộng

- 1 Thăm nút  $v_s$ .
- 2 Tìm  $VA$  là tập các nút kề của  $v_s$  mà chưa được thăm.
- 3 Thăm lần lượt các nút trong  $VA$ .
- 4 Lặp lại giải thuật cho từng nút trong  $VA$ .

# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

## Lưu ý:

- Để cài đặt giải thuật trên, ta cần một cấu trúc dữ liệu trung gian để lưu các nút kề mà sau này sẽ lần lượt được thăm. Có thể thấy danh sách kiểu *hàng đợi* là cấu trúc phù hợp vì các nút được lưu trước sẽ được thăm trước.
- Gọi hàng đợi cần dùng là  $Q$ , khi đó gọi thủ tục duyệt theo chiều rộng là  $BreathFirstSearch(G, v_s)$  mà sẽ được cài đặt như sau.

# Phép duyệt đồ thị - Duyệt theo chiều rộng

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

```
void BreathFirstSearch(G,  $v_s$ )
```

```
{  
    insert( $v_s$ , Q);  
    while (!isEmpty(Q))  
    {  
        u = remove(Q);  
        visit(u);  
        for each  $u_a$ : (unvisited) and  
                     (adjacent to u) and ( $u_a \notin Q$ )  
            insert( $u_a$ , Q);  
    }  
}
```

# Duyệt theo chiều rộng - Thủ tục $BreadthFirstSearch(G, v_s)$

Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị

## Lưu ý

Ta có thể nhận thấy độ phức tạp của thủ tục trên phụ thuộc chủ yếu vào thao tác tìm kiếm các nút kề  $u_a$ . Có một kỹ thuật cài đặt giúp giảm độ phức tạp của thao tác trên là bổ sung thêm một trạng thái nữa cho các nút, thay vì chỉ có hai trạng thái là *đã được thăm*/*chưa được thăm* như ở giải thuật duyệt theo chiều sâu. Ba trạng thái này thường được biểu diễn ngắn gọn bằng ba màu trắng, xám, và đen cho mỗi nút:

- *Nút trắng*: là nút chưa được thăm và chưa nằm trong hàng đợi.
- *Nút xám*: là nút chưa được thăm nhưng đang nằm trong hàng đợi.
- *Nút đen*: là nút đã được thăm.

# Duyệt theo chiều rộng - Ví dụ

Cấu trúc Đồ  
thị

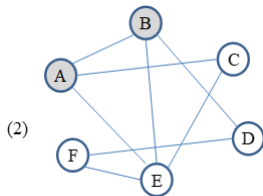
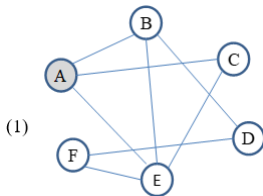
Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

Các hình dưới đây minh họa hoạt động của giải thuật duyệt theo chiều rộng. Để đơn giản, mỗi nút chỉ có 2 trạng thái *đã thăm-chưa thăm*, tương ứng với hai màu *đen-trắng*.



# Duyệt theo chiều rộng - Ví dụ

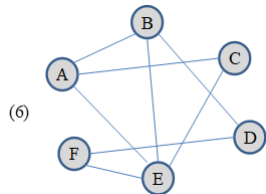
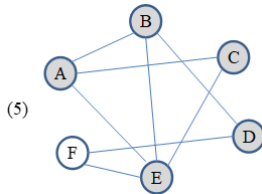
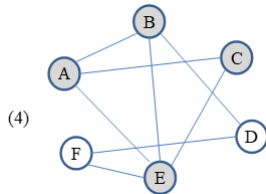
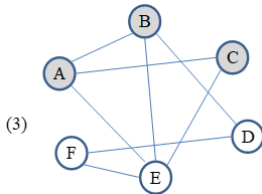
Cấu trúc Đồ thị

Nguyễn Thanh Bình

Cơ bản về đồ thị

Cài đặt đồ thị

Phép duyệt đồ thị



# Tham khảo

Cấu trúc Đồ  
thị

Nguyễn  
Thanh Bình

Cơ bản về đồ  
thị

Cài đặt đồ thị

Phép duyệt đồ  
thị

- “Ngôn ngữ lập trình C và Cấu trúc dữ liệu”; Nguyễn Thanh Bình, Nguyễn Hoài Giang; NXB Giáo Dục Việt Nam, 2017.