

## **BÀI 11**

### **Chương 6** **Các thuật toán duyệt đồ thị**

Phép duyệt đồ thị là một cách liệt kê tất cả các đỉnh của đồ thị này thành một danh sách tuyến tính. Hay nói một cách khác, phép duyệt đồ thị cho ta một cách “*đi qua*” tất cả các đỉnh của đồ thị để truy nhập, thêm bớt thông tin ở các đỉnh của đồ thị đó.

Phép duyệt đồ thị không phụ thuộc vào hướng của các cạnh. Do vậy, với đồ thị có hướng thì ta vô hướng hoá trước khi duyệt.

#### 6.1. Các thuật toán duyệt đồ thị

##### 6.1.1. *Thuật toán duyệt đồ thị*

Giả sử  $G = (V, E)$  là đồ thị đã cho và  $x_0$  là một đỉnh nào đó của  $G$ . Ký hiệu DS là một cấu trúc dữ liệu kiểu danh sách dùng để chứa các đỉnh.

- 1) Khởi đầu:  $DS \leftarrow \{x_0\}$
- 2) Lấy đỉnh  $x$  ra khỏi đầu DS
- 3) Duyệt đỉnh  $x$
- 4) Nạp các đỉnh của  $F(x)$  vào DS
- 5) Nếu  $DS \neq \emptyset$  thì quay lên bước 2)
- 6) Dừng

##### 6.1.2. *Duyệt theo chiều sâu* (Depth-First Search)

Nếu trong thuật toán trên, danh sách DS được tổ chức theo kiểu stack (*danh sách vào sau - ra trước – LIFO*) thì ta có phương pháp duyệt theo chiều sâu.

Trong phương pháp này mỗi lần duyệt một đỉnh ta duyệt đến tận cùng mỗi nhánh rồi mới chuyển sang duyệt nhánh khác.

Giả sử  $G = (V, E)$  là một đồ thị vô hướng.

Ta bắt đầu duyệt từ một đỉnh  $x_0$  nào đó của đồ thị. Sau đó chọn  $x$  là đỉnh kề nào đó của  $x_0$  và lặp lại quá trình duyệt đối với đỉnh  $x$ .

Giả sử ta đang xét đỉnh  $x$ . Nếu trong số các đỉnh kề với  $x$ , ta tìm được đỉnh  $y$  chưa được duyệt thì ta sẽ xét đỉnh này và bắt đầu từ đó ta tiếp tục quá trình duyệt.

Ngược lại, nếu không còn đỉnh nào kề với  $x$  chưa được duyệt thì ta nói rằng đỉnh  $x$  đã duyệt xong và quay trở lại tiếp tục duyệt từ đỉnh mà từ đó ta đến được đỉnh  $x$ . Nếu quay trở lại đúng đỉnh  $x_0$  thì phép duyệt kết thúc.

**Thuật toán 6.1** (Duyệt đồ thị theo chiều sâu):

*Dữ liệu:* Biểu diễn mảng DK các danh sách kề của đồ thị vô hướng  $G$ .

*Kết quả:* Danh sách các đỉnh của đồ thị  $G$ .

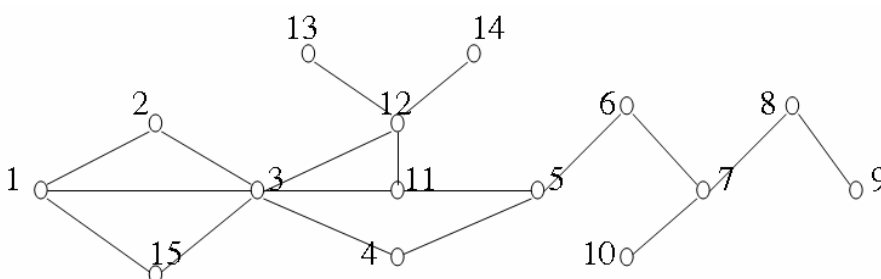
```

1  procedure D_SAU ( $v$ ) ;
2  begin
3      Thăm_đỉnh ( $v$ ) ;
4      Duyệt [ $v$ ] := true ;
5      for  $u \in DK[v]$  do
6          if ! Duyệt [ $u$ ] then D_SAU ( $u$ ) ;
7  end ;

8  BEGIN                                     { Chương trình chính }
9      for  $v \in V$  do Duyệt [ $v$ ] := false ;
10     for  $v \in V$  do
11         if ! Duyệt [ $v$ ] then D_SAU ( $v$ ) ;
12 END.
```

Độ phức tạp của thuật toán là:  $O(n+m)$

Ví dụ 6.1: Đồ thị được duyệt theo chiều sâu.



Hình 6.1. Thứ tự của các đỉnh được duyệt theo chiều sâu

Trong thuật toán duyệt theo chiều sâu, đỉnh được thăm càng muộn càng sớm trở thành duyệt xong. Do vậy việc dùng một ngăn xếp (*stack*) để lưu trữ các đỉnh đang duyệt là rất thích hợp. Ta có thủ tục cải tiến sau đây:

**Thuật toán 6.2** (Duyệt đồ thị theo chiều sâu):

```

1  procedure D_SAU_2 (v) ;
2  begin
3      S :=  $\emptyset$  ;
4      Thăm_đỉnh (v) ;
5      Duyệt [v] := true ;
6      push v onto S ;           { Nạp v lên đỉnh của S }
7      while S  $\neq \emptyset$  do
8          begin
9              while  $\exists u \in \text{DK}[\text{top}(S)]$  do
10                 if ! Duyệt [u] then
11                     begin
12                         Thăm_đỉnh (u) ;
13                         Duyệt [u] := true ;
14                         push u onto S ;
15                     end ;
16                 pop(S) ;        { Loại bỏ phần tử ở đỉnh của S }
17             end ;
18 end ;

```