

## **BÀI 12**

### **6.1.3. Duyệt theo chiều rộng** (Breadth-First Search)

Nếu trong thuật toán duyệt đồ thị, cấu trúc danh sách DS được tổ chức theo kiểu hàng đợi (*danh sách vào trước - ra trước – FIFO* ) thì ta có phương pháp duyệt theo chiều rộng. Trong phương pháp này việc duyệt có tính chất “lan rộng”. Một đỉnh được duyệt xong ngay sau khi ta đã xét hết tất cả các đỉnh kề với nó. Đỉnh được xét càng sớm thì sớm trở thành duyệt xong.

**Thuật toán 6.3** (Duyệt đồ thị theo chiều rộng):

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

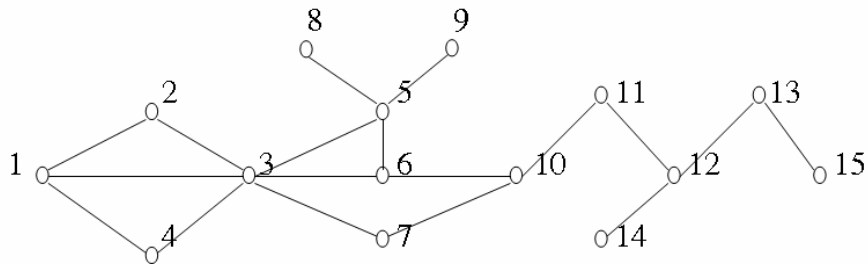
*Kết quả:* Danh sách các đỉnh của đồ thị G.

```
1  procedure D_RONG (v) ;
2  begin
3      Q :=  $\emptyset$  ;
4      enqueue v into Q ;           { Nạp v vào cuối hàng đợi Q }
5      Duyệt [v] := true ;
6      while Q  $\neq \emptyset$  do
7          begin
8              dequeue z from Q ;     { Loại z ra khỏi đầu hàng đợi Q }
9              Thăm_đỉnh (z) ;
10             for u  $\in$  DK[z] do
11                 if ! Duyệt [u] then
12                     begin
13                         enqueue u into Q ;
14                         Duyệt [u] := true ;
15                     end ;
16             end ;
17 end ;

18 BEGIN                               { Chương trình chính }
19     for v  $\in$  V do Duyệt [v] := false ;
20     for v  $\in$  V do
21         if ! Duyệt [v] then D_RONG (v) ;
22 END .
```

Thuật toán này cũng có độ phức tạp là:  $O(n+m)$

Ví dụ 6.2: Đồ thị trong Ví dụ 6.1 được duyệt theo chiều rộng.



Hình 6.2. Thứ tự của các đỉnh được duyệt theo chiều rộng

## 6.2. Một số ứng dụng của thuật toán duyệt đồ thị

### 6.2.1. Bài toán tìm đường đi

Giả sử  $a$  và  $b$  là hai đỉnh nào đó của đồ thị vô hướng  $G$ . Hãy tìm đường đi (nếu có) từ đỉnh  $a$  đến đỉnh  $b$ .

Bài toán này đã được giải quyết bằng Thuật toán 1.4 hoặc thuật toán Warshall. Theo các phương pháp duyệt đồ thị, các lời gọi thủ tục  $D\_SAU(a)$  hoặc  $D\_RONG(a)$  cho phép thăm tất cả các đỉnh thuộc cùng thành phần liên thông với đỉnh  $a$ . Vì vậy, sau khi thực hiện xong thủ tục nếu  $Duyet[b] = \text{false}$  thì không có đường đi từ đỉnh  $a$  đến đỉnh  $b$  ngược lại thì đỉnh  $b$  thuộc cùng mảng liên thông với đỉnh  $a$ , hay nói một cách khác, có đường đi từ  $a$  đến  $b$ .

Để khôi phục đường đi ta dùng thêm một biến mảng  $Truoc$ . Thành phần  $Truoc[u]$  ghi lại đỉnh đến trước đỉnh  $u$  trên đường duyệt từ  $a$  tới  $u$ . Khi đó, trong thủ tục  $D\_SAU(v)$  cần sửa đổi câu lệnh `if` ở dòng lệnh 6 như sau:

```
6      if !Duyet[u] then begin Truoc[u] := v ; D_SAU(u) end ;
```

còn trong thủ tục  $D\_RONG(v)$  thì cần sửa câu lệnh `if` ở các dòng lệnh 11-15 là:

```
11     if !Duyet[u] then
12         begin
13             enqueue u into Q ;
14             Duyet[u] := true ; Truoc[u] := z
15         end ;
```

Đường đi cần tìm (nếu có) sẽ được khôi phục như sau:

$$b \leftarrow a_1 = Truoc[b] \leftarrow a_2 = Truoc[a_1] \leftarrow \dots \leftarrow a$$

Chú ý: Đường đi tìm được theo thuật toán duyệt theo chiều rộng là đường đi ngắn nhất từ đỉnh  $a$  đến đỉnh  $b$ .

### 6.2.2. Bài toán tìm các mảng liên thông

Cho đồ thị  $G$ , hãy tìm số mảng liên thông  $p$  của đồ thị này và xác định xem mỗi mảng liên thông bao gồm những đỉnh nào.

Do các thủ tục  $D\_SAU(v)$  hoặc  $D\_RONG(v)$  cho phép duyệt tất cả các đỉnh thuộc cùng mảng liên thông với đỉnh  $v$  nên số mảng liên thông  $p$  của đồ thị  $G$  chính bằng số lần gọi đến các thủ tục này trong chương trình chính.

Để ghi nhận các đỉnh trong từng mảng liên thông, ta dùng thêm biến mảng  $Mang[v]$  để ghi chỉ số của mảng liên thông chứa đỉnh  $v$ . Chỉ số này tăng từ 1 đến  $p$ .

Ta dùng biến  $p$  để đếm số mảng liên thông của đồ thị và gán chỉ số cho các mảng liên thông tìm được. Bắt đầu nó được khởi tạo giá trị bằng 0. Thủ tục  $Thăm\_đỉnh(v)$  trong các thủ tục  $D\_SAU(v)$  và  $D\_RONG(v)$  làm thêm nhiệm vụ gán:

$$Mang[v] := p;$$

Còn chương trình chính của thuật toán duyệt cần được sửa lại như sau:

```
1 BEGIN                                     { Chương trình chính }
2   for  $v \in V$  do Duyet [ $v$ ] := false ;
3    $p := 0$  ;
4   for  $v \in V$  do
5     if ! Duyet [ $v$ ] then
6       begin  $p := p + 1$  ;
7              $D\_SAU(v)$  ;           {  $D\_RONG(v)$  ; }
8       end ;
9 END .
```

Khi chương trình kết thúc, biến  $p$  sẽ cho số mảng liên thông của đồ thị còn các giá trị của biến mảng  $Mang[v]$ ,  $v \in V$  cho phép liệt kê tất cả các đỉnh trong từng mảng liên thông của đồ thị.