

Bài Toán 13: Thuật Toán DFS Trên Đồ Thị Tổng Quát

Đồ Án 5.2: Depth-first Search

Phát biểu bài toán

Cho một đồ thị tổng quát $G = (V, E)$, không giới hạn kiểu đồ thị. Yêu cầu: Viết thuật toán DFS để duyệt toàn bộ đồ thị, bao gồm cả các thành phần rời rạc, cạnh trùng, cạnh tự khép và đồ thị có hướng hoặc vô hướng.

Tính chất đồ thị tổng quát

- Có thể là đồ thị vô hướng hoặc hữu hướng
- Cho phép nhiều cạnh giữa cùng một cặp đỉnh (đa cung)
- Cho phép self-loop (u, u)
- Có thể gồm nhiều thành phần liên thông
- Có thể rỗng

Ý tưởng thuật toán

- Khởi tạo mảng `visited[]` cho tất cả các đỉnh
- Duyệt DFS từ từng đỉnh chưa thăm \rightarrow đảm bảo bao phủ toàn bộ đồ thị (gồm nhiều thành phần)
- Mỗi đỉnh chỉ được duyệt đúng một lần, bất kể có bao nhiêu cạnh đi tới nó

Thuật toán DFS tổng quát (pseudocode)

```
DFS_Component(G, u):  
    visited[u]  $\leftarrow$  True  
    xử lý u  
    for mỗi v  $\in$  adj[u]:  
        if not visited[v]:  
            DFS_Component(G, v)
```

```

General_DFS(G):
    visited[v] ← False với mọi v ∈ V
    for mỗi đỉnh u:
        if not visited[u]:
            DFS_Component(G, u)

```

Chú thích các biến số

- **G**: đồ thị tổng quát (đa cung, self-loop, rời rạc)
- **adj[u]**: danh sách các đỉnh kề với u (có thể trùng)
- **visited[u]**: true nếu đã thăm đỉnh u
- **DFS_Component**: hàm xử lý một thành phần liên thông
- **General_DFS**: vòng lặp chính để bao phủ toàn bộ đồ thị

Xử lý các trường hợp đặc biệt

- **Cạnh trùng (multiedge)**: **visited[]** đảm bảo duyệt đúng 1 lần
- **Self-loop**: DFS không gọi lại chính nó nếu đã được đánh dấu
- **Disconnected graph**: Mỗi thành phần liên thông được duyệt riêng