

# Bài Toán 12: Thuật Toán DFS Trên Multigraph (Đồ Thị Đa Cung)

## Đồ Án 5.2: Depth-first Search

### Phát biểu bài toán

Cho một đồ thị đa cung  $G = (V, E)$ , trong đó có thể tồn tại nhiều cạnh giữa cùng một cặp đỉnh  $(u, v)$ . Yêu cầu: Triển khai thuật toán tìm kiếm theo chiều sâu (Depth-First Search – DFS) trên  $G$ .

### Đặc điểm của multigraph

- Có thể tồn tại nhiều cạnh giữa một cặp đỉnh
- Có thể tồn tại cạnh tự khép  $(u, u)$
- Danh sách kề có thể chứa trùng lặp các đỉnh

### Ý tưởng thuật toán

- DFS được triển khai tương tự như với đồ thị đơn
- Mỗi đỉnh của một đồ thị được duyệt đúng một lần, điều này vẫn đúng bất kể đồ thị có chứa các cạnh bội (cạnh trùng) hay không.
- Dùng mảng `visited[]` để ngăn việc lặp lại duyệt đỉnh

### Thuật toán DFS (pseudocode)

```
DFS_Multigraph(G, u):  
    visited[u] ← True  
    xử lý đỉnh u  
    for mỗi v  adj[u]: // có thể trùng  
        if not visited[v]:  
            DFS_Multigraph(G, v)
```

## Chú thích các biến số

- $G$ : đồ thị đa cung, biểu diễn dưới dạng danh sách kề (có thể chứa trùng)
- $adj[u]$ : danh sách các đỉnh kề với  $u$  (có thể có lặp)
- $visited[v]$ : boolean kiểm tra đỉnh  $v$  đã được duyệt chưa
- $u$ : đỉnh hiện tại trong DFS
- $res$ : danh sách các đỉnh được duyệt theo thứ tự

## Xử lý cạnh trùng và self-loop

- Cạnh trùng: nếu  $adj[u] = [v, v, v]$  thì DFS vẫn chỉ gọi một lần cho  $v$
- Cạnh tự khép  $(u, u)$ : sẽ không gây ra vòng lặp vô hạn nếu kiểm tra  $visited[u]$  đúng