

# Bài Toán 10: Thuật Toán BFS Trên Đồ Thị Tổng Quát

## Đồ Án 5.1: Breadth-first Search

### Phát biểu bài toán

Cho một đồ thị tổng quát  $G = (V, E)$ , không giả định đơn, đa cung hay hướng/vô hướng. Viết thuật toán Breadth-First Search (BFS) có khả năng hoạt động chính xác trên mọi loại đồ thị.

### Giả thuyết

- Đồ thị có thể có nhiều thành phần liên thông
- Đỉnh có thể có self-loop (tức là  $(v, v) \in E$ )
- Có thể có nhiều cạnh nối giữa một cặp đỉnh
- Có thể là đồ thị hữu hướng hoặc vô hướng
- Có thể rỗng

### Ý tưởng

- Sử dụng BFS tiêu chuẩn cho từng thành phần liên thông
- Duyệt BFS từ mọi đỉnh chưa được thăm để bao phủ toàn bộ đồ thị
- Sử dụng mảng `visited[]` để tránh lặp chu kỳ, đa cung hoặc self-loop

### Thuật toán BFS tổng quát (pseudocode)

`General_BFS(G):`

```
n ← số lượng đỉnh
visited[v] ← False  v ∈ V
for mỗi đỉnh u từ 0 đến n-1:
    if not visited[u]:
        Q ← hàng đợi mới
        enqueue(Q, u)
```

```

visited[u] ← True
while Q không rỗng:
    v ← dequeue(Q)
    xử lý v
    for mỗi đỉnh w ∈ adj[v]:
        if not visited[w]:
            visited[w] ← True
            enqueue(Q, w)

```

## Chú thích các biến số

- $G$ : đồ thị tổng quát, có thể có hướng, đa cung, self-loop
- $\text{adj}[v]$ : danh sách các đỉnh kề với  $v$  (có thể trùng)
- $\text{visited}[v]$ : boolean kiểm tra đã duyệt đỉnh  $v$
- $Q$ : hàng đợi BFS dùng cho từng thành phần liên thông
- $\text{res}$ : danh sách kết quả BFS từ từng component

## Xử lý bài toán

- Với self-loop:  $(v, v)$  không ảnh hưởng nếu đã kiểm tra  $\text{visited}[v]$  đúng cách
- Với multigraph: dù có nhiều cạnh trùng, BFS chỉ duyệt 1 lần
- Với đồ thị không liên thông: dùng BFS trên từng thành phần