

Bài Toán 9: Thuật Toán BFS Trên Multigraph (Đồ Thị Đa Cung)

Đồ Án 5.1: Breadth-first Search

Phát biểu bài toán

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

Đặc điểm của multigraph

- Có thể có nhiều hơn một cạnh giữa 2 đỉnh u và v
- Có thể có cạnh tự khép (self-loop), ví dụ (u, u)
- Danh sách kề có thể chứa nhiều lần cùng một đỉnh kề

Ý tưởng

- Duyệt theo chiều rộng như đồ thị đơn
- Để tránh duyệt lặp qua nhiều cạnh trùng nhau, cần dùng mảng `visited[]` để đánh dấu đã thăm
- Mỗi đỉnh chỉ được duyệt đúng một lần, bỏ qua các cạnh trùng nếu đỉnh kề đã thăm

Thuật toán BFS (pseudocode)

```
BFS_Multigraph(G, s):  
    visited[v] ← False với mọi v ∈ V  
    Q ← hàng đợi rỗng  
    visited[s] ← True  
    enqueue(Q, s)  
    while Q không rỗng:  
        u ← dequeue(Q)
```

```

xử lý đỉnh u
for mỗi v  adj[u]: // cho phép lặp
    if not visited[v]:
        visited[v] ← True
        enqueue(Q, v)

```

Chú thích các biến số

- G : đồ thị đa cung, biểu diễn bằng danh sách kề có thể chứa trùng
- $adj[u]$: danh sách các đỉnh kề với u , có thể chứa v nhiều lần
- s : đỉnh bắt đầu BFS
- $visited[v]$: boolean đánh dấu đỉnh v đã được duyệt
- Q : hàng đợi FIFO dùng để duyệt BFS
- res : danh sách kết quả duyệt

Giải thích xử lý trùng cạnh

- Tại vì một đỉnh v có thể xuất hiện nhiều lần trong $adj[u]$, ta chỉ xét $visited[v]$ đúng một lần
- Ví dụ: nếu $adj[0] = [1, 1, 2]$, thì đỉnh 1 chỉ được duyệt 1 lần duy nhất
- Điều này sẽ giúp BFS vẫn có độ phức tạp $O(V + E)$ với E là tổng số cung (bao gồm trùng cạnh)