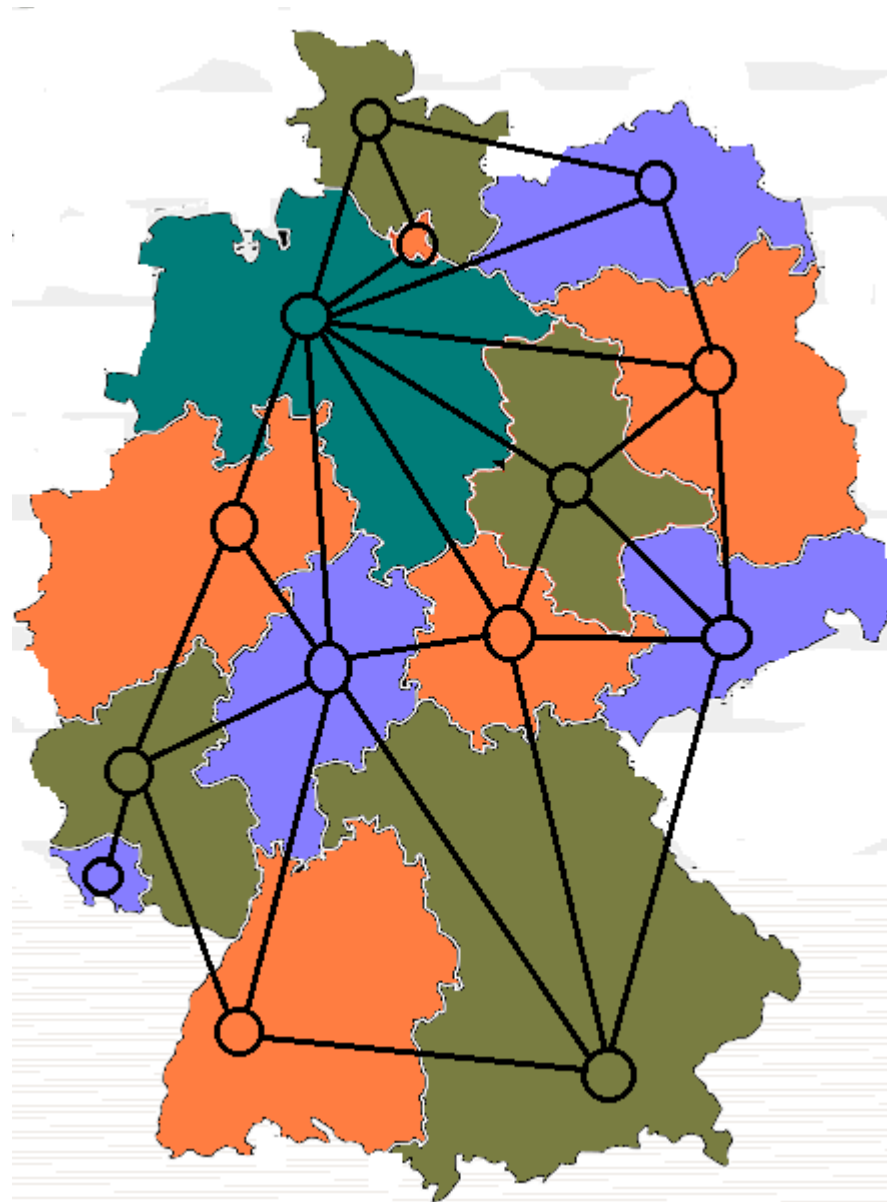


CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ VÀ ỨNG DỤNG

Vũ Ngọc Thanh Sang



CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ VÀ ỨNG DỤNG

1. Tìm kiếm theo chiều sâu
2. Tìm kiếm theo chiều rộng
3. Một số ứng dụng

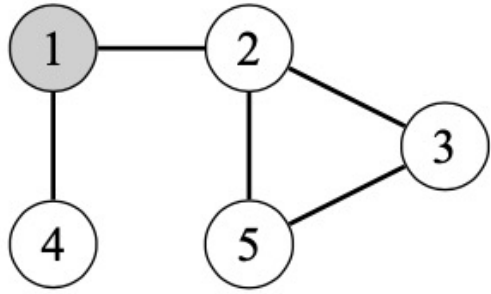
CÁC THUẬT TOÁN TÌM KIẾM TRÊN ĐỒ THỊ VÀ ỨNG DỤNG

- Có hai cách để duyệt đồ thị đó là:
 - **Duyệt theo chiều sâu** (depth-first search)
 - **Duyệt theo chiều rộng** (breadth-first search)
- Cả hai đều bắt đầu từ một đỉnh cho trước và kết thúc khi tất cả các đỉnh được duyệt. Điểm khác nhau là thứ tự các đỉnh được duyệt.

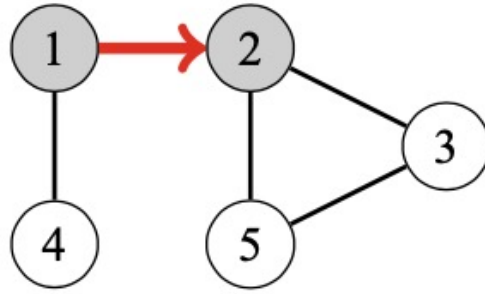
TÌM KIẾM THEO CHIỀU SÂU (Depth-First Search-DFS)

- Tìm kiếm theo chiều sâu (DFS) là một kỹ thuật duyệt đồ thị đơn giản.
- Thuật toán bắt đầu tại một đỉnh và tất cả các đỉnh khác liên thông trực tiếp với đỉnh bắt đầu.
- DFS luôn đi theo một đường dẫn duy nhất miễn là nó tìm thấy các đỉnh mới. Sau đó nó quay trở lại các đỉnh trước và bắt đầu tìm kiếm đến các phần khác của đồ thị.
- Thuật toán chỉ truy cập và xử lý **mỗi đỉnh một lần**.

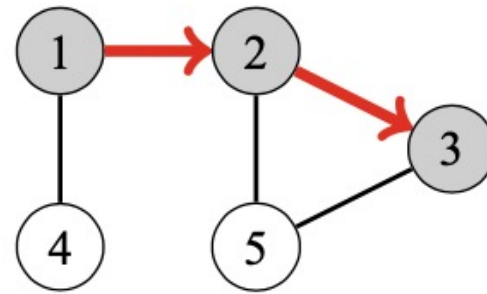
THUẬT TOÁN DFS



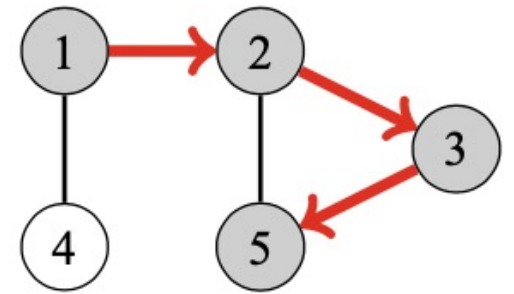
step 1



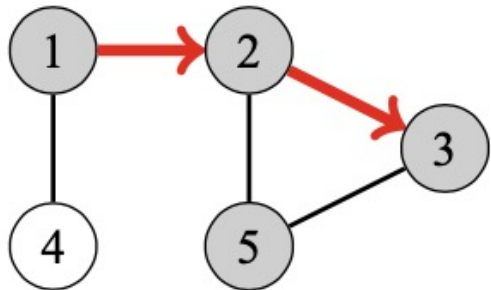
step 2



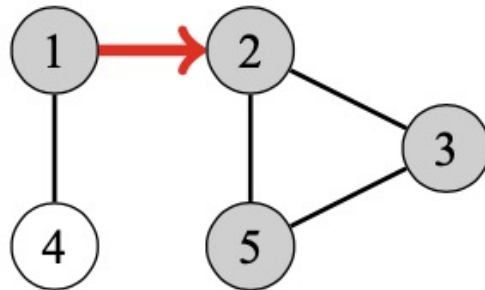
step 3



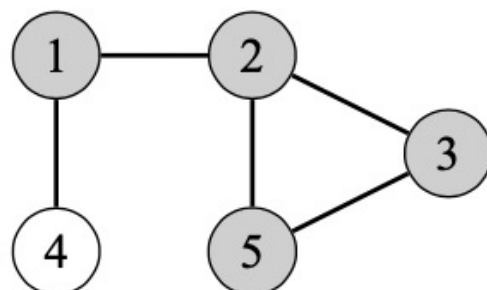
step 4



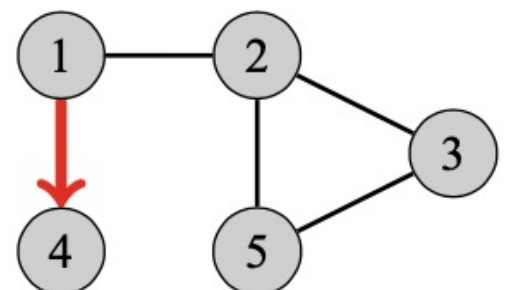
step 5



step 6



step 7



step 8

THUẬT TOÁN DFS

- Giả sử đồ thị được biểu diễn bằng danh sách kề cận

```
vector<int> adj[N]; // adjacency lists
bool visited[N]; // tracking visited vertices

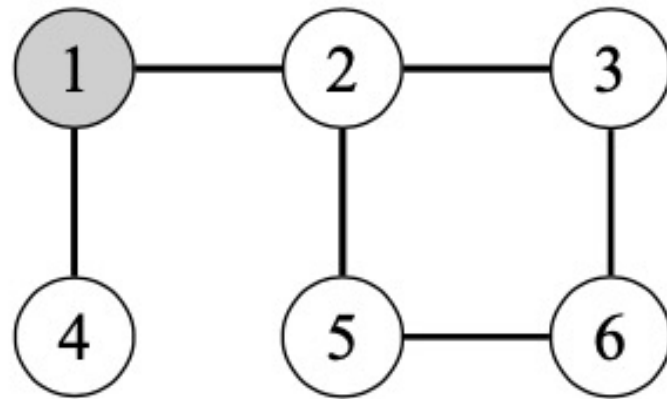
void dfs(int s) {
    if (visited[s])
        return; // already processed vertices
    visited[s] = true;
    // process node s
    for (auto u: adj[s]) {
        dfs(u);
    }
}
```

- Độ phức tạp của thuật toán tìm kiếm theo chiều sâu là $O(n + m)$ với n là số đỉnh và m là số cạnh. Thuật toán duyệt qua mỗi đỉnh và cạnh một lần

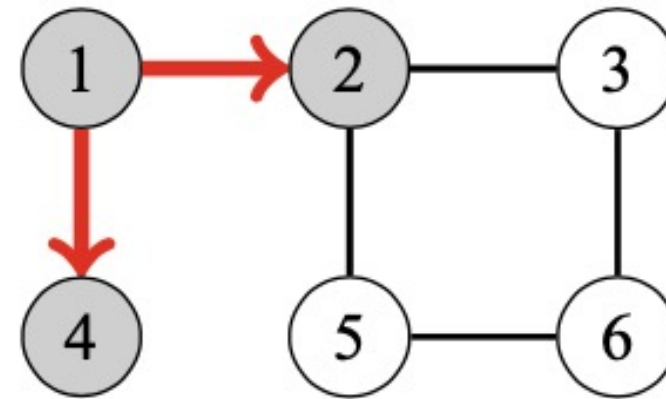
TÌM KIẾM THEO CHIỀU RỘNG (Breadth-First Search-BFS)

- Tìm kiếm theo chiều rộng (BFS) truy cập các đỉnh của đồ thị theo thứ tự tăng dần khoảng cách so với đỉnh bắt đầu.
- Khi bắt đầu thuật toán tìm kiếm các đỉnh có khoảng cách tới đỉnh bắt đầu là 1, sau đó là các đỉnh có khoảng cách là 2. Quá trình này được tiếp tục cho đến khi tất cả các đỉnh đã được duyệt.
- Tìm kiếm theo chiều rộng khó thực hiện hơn tìm kiếm theo chiều sâu, vì thuật toán truy cập các đỉnh ở các phần khác nhau của đồ thị.

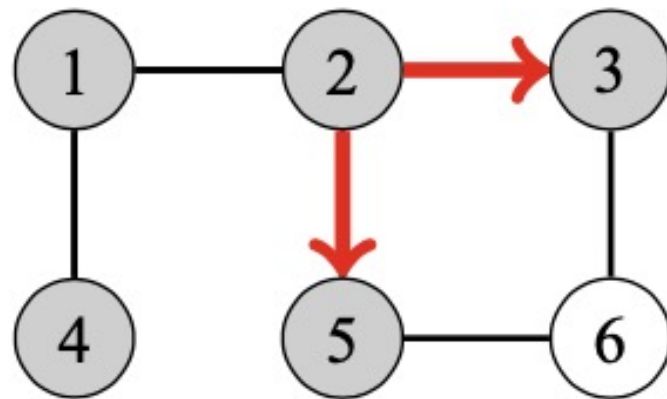
THUẬT TOÁN BFS



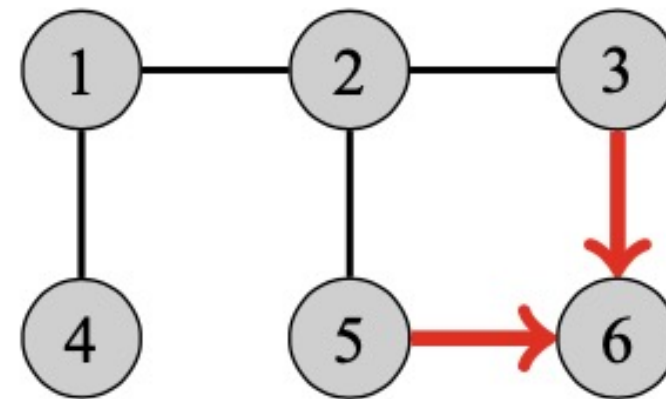
step 1



step 2



step 3



step 4

THUẬT TOÁN BFS

- Giả sử đồ thị được biểu diễn bằng danh sách kề cận.
- Để duyệt theo chiều rộng, ta tổ sử dụng hàng đợi (queue). Tại mỗi giai đoạn, đỉnh tiếp theo của hàng đợi sẽ được xử lý.

```
queue<int> q;
```

```
bool visited[N];
```

```
int distance[N];
```

- Hàng đợi q chứa các đỉnh được xử lý theo thứ tự khoảng cách của chúng tăng dần. Các đỉnh mới luôn được thêm vào cuối hàng đợi và đỉnh ở đầu hàng đợi là đỉnh tiếp theo được xử lý.

THUẬT TOÁN BFS

- Việc tìm kiếm có thể được thực thi như sau, bắt đầu từ đỉnh x:

```
visited[x] = true;
distance[x] = 0;
q.push(x);
while (!q.empty()) {
    int s = q.front();
    q.pop();
    // process node s
    for (auto u : adj[s]) {
        if (visited[u]) continue;
        visited[u] = true;
        distance[u] = distance[s]+1;
        q.push(u);
    }
}
```

- Độ phức tạp của thuật toán tìm kiếm theo chiều rộng là $O(n + m)$ với n là số đỉnh và m là số cạnh. Thuật toán duyệt qua mỗi đỉnh và cạnh một lần

MỘT SỐ ỨNG DỤNG

- Tìm đường đi giữa hai đỉnh
- Tính số thành phần liên thông
- Kiểm tra tính liên thông

THANK YOU FOR YOUR ATTENTION.