

Thuật toán duyệt đồ thị

Bài toán

- Cho trước đồ thị G và một điểm v
- Tìm tất cả các điểm có thể đi đến từ v (tức là tồn tại đường đi từ v đến điểm đó)
 - Thực hiện tác vụ gì đó như in ra thông tin, thiết lập giá trị...

Phương pháp duyệt đồ thị

Phương pháp cơ bản để duyệt đồ thị:

- Chọn một điểm làm điểm bắt đầu
- Tạo ra tập điểm kề với điểm hiện tại
- Thăm từng đỉnh trong tập và “đánh dấu” đỉnh đó sau khi thăm để không thăm lại
- Lặp lại quy trình trên (bỏ qua các đỉnh được đánh dấu đã thăm) đến khi không còn đỉnh nào.

Giả mã

```
traverseGraph(Node start) {
    Set pending = emptySet();
    pending.add(start)
    mark start as visited
    while(pending is not empty) {
        next = pending.remove()
        for each node u adjacent to next
            if(u is not marked) {
                mark u
                pending.add(u)
            }
        }
    }
}
```

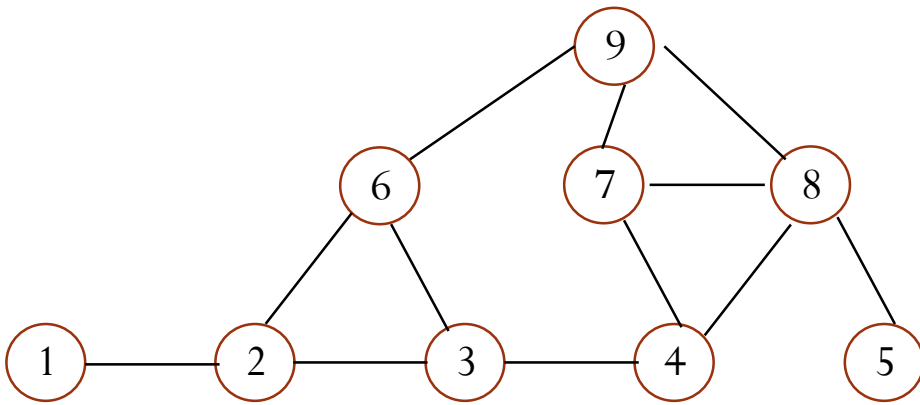
Thứ tự duyệt

Thứ tự duyệt đồ thị phụ thuộc hoàn toàn vào cách thêm đỉnh mới và lựa chọn đỉnh được thăm:

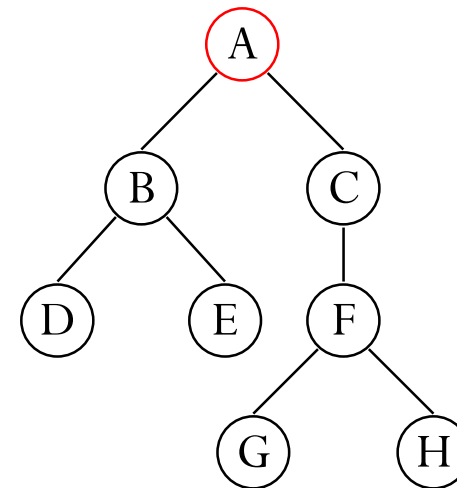
- DFS - depth-first graph search: tìm kiếm trên đồ thị theo chiều sâu – sử dụng ngăn xếp (stack)
- BFS - breadth-first graph search: tìm kiếm trên đồ thị theo chiều rộng – sử dụng hàng đợi (queue)

Duyệt đồ thị theo chiều sâu

- Ý tưởng:
 - Đi xa nhất theo từng nhánh
 - Khi duyệt hết một nhánh, lùi về từng đỉnh để tìm và duyệt nhánh tiếp theo
 - Dừng lại khi thấy đỉnh cần tìm hoặc duyệt hết



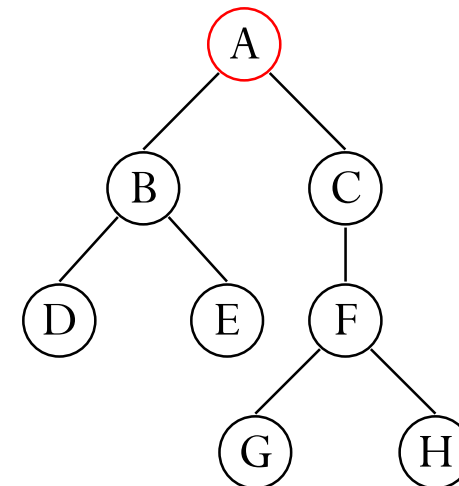
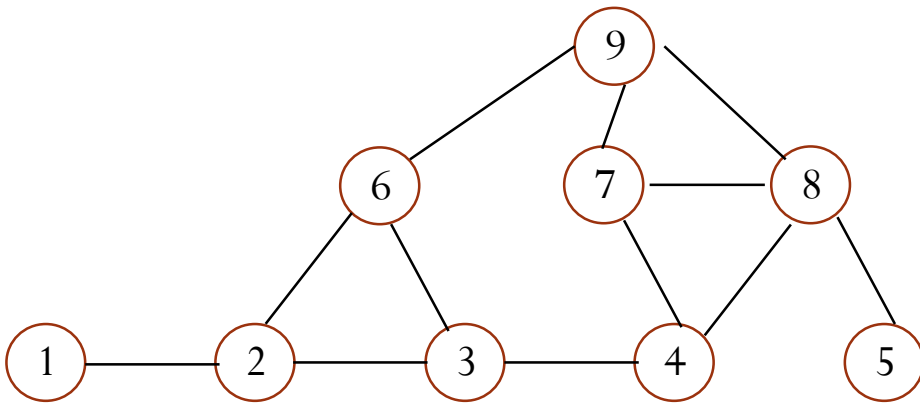
1 - 2 - 3 - 4 - 7 - 8 - 5 - 9 - 6



Duyệt đồ thị theo chiều rộng

- Ý tưởng:
 - Duyệt tất cả các đỉnh gần đỉnh hiện tại trước

1 - 2 - 3 - 6 - 4 - 9 - 7 - 8 - 5



So sánh

- Tìm kiếm theo chiều rộng luôn tìm ra đường đi ngắn nhất từ một đỉnh bắt đầu đến một đỉnh đích
- Độ phức tạp thời gian như nhau
- Độ phức tạp không gian:
 - Không gian cần dùng cho thuật toán tìm kiếm theo chiều sâu có thể rất lớn
 - Tìm kiếm theo chiều sâu tốn ít không gian hơn trong quá trình tìm kiếm