

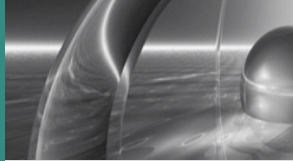
## CHƯƠNG 3

# TÌM KIẾM TRÊN ĐỒ THỊ

# Nội dung

- 1 Bài toán tìm kiếm trên đồ thị.....●
- 2 Depth First Search (DFS).....●
- 3 Breadth First Search (BFS).....●
- 4 Thảo luận & Bài tập.....●

# Bài toán tìm kiếm trên đồ thị (1/2)



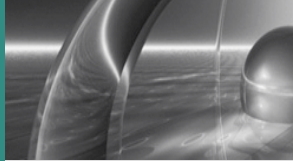
## Tìm đường đi từ một đỉnh đến một đỉnh khác

- Kiểm tra tính liên thông
- Kiểm tra tính liên thông mạnh
- Xác định các thành phần liên thông của đồ thị

## Duyệt qua các đỉnh của đồ thị

- Cập nhật, xử lý dữ liệu tại các đỉnh của đồ thị

# Bài toán tìm kiếm trên đồ thị (2/2)



Tìm đường đi:

- Từ đỉnh xuất phát  $s$
- Đến đỉnh đích  $t$

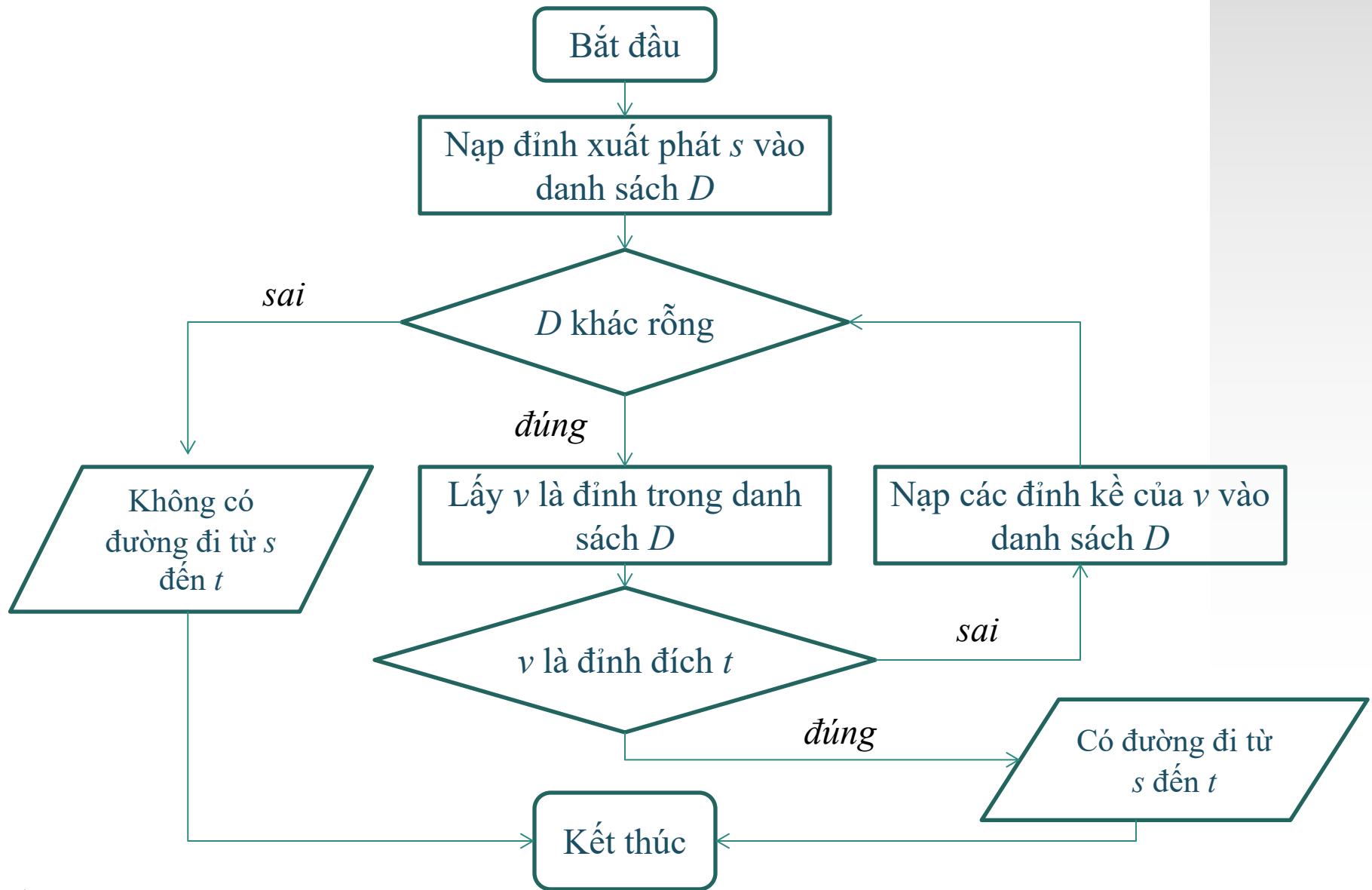
Yêu cầu 1:

- Tồn tại đường đi
- Hay không tồn tại đường đi

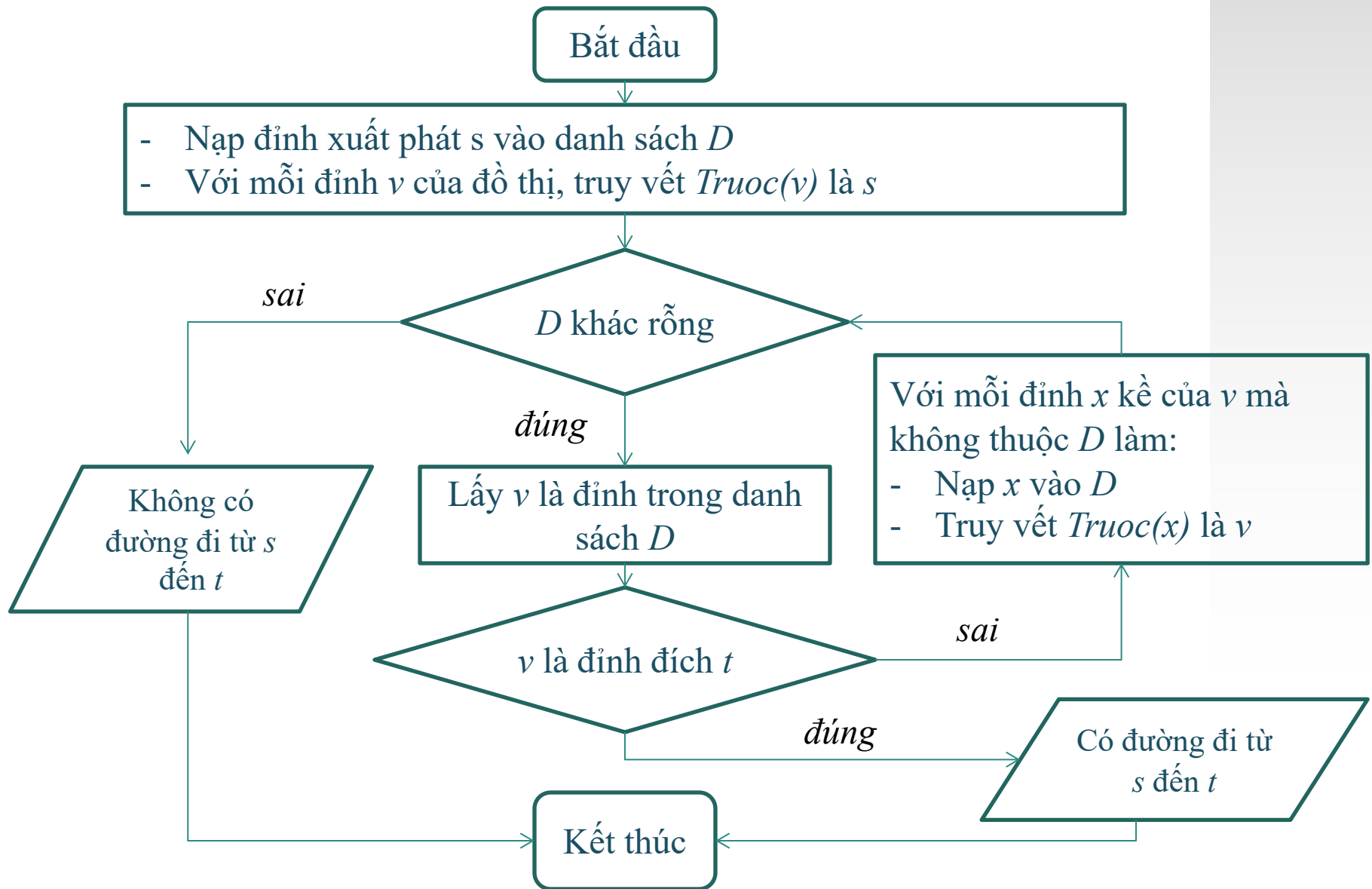
Yêu cầu 2:

- Nếu tồn tại đường đi từ  $s \rightarrow t$  thì đi như thế nào?

# Thuật toán cho yêu cầu 1



# Thuật toán cho yêu cầu 2

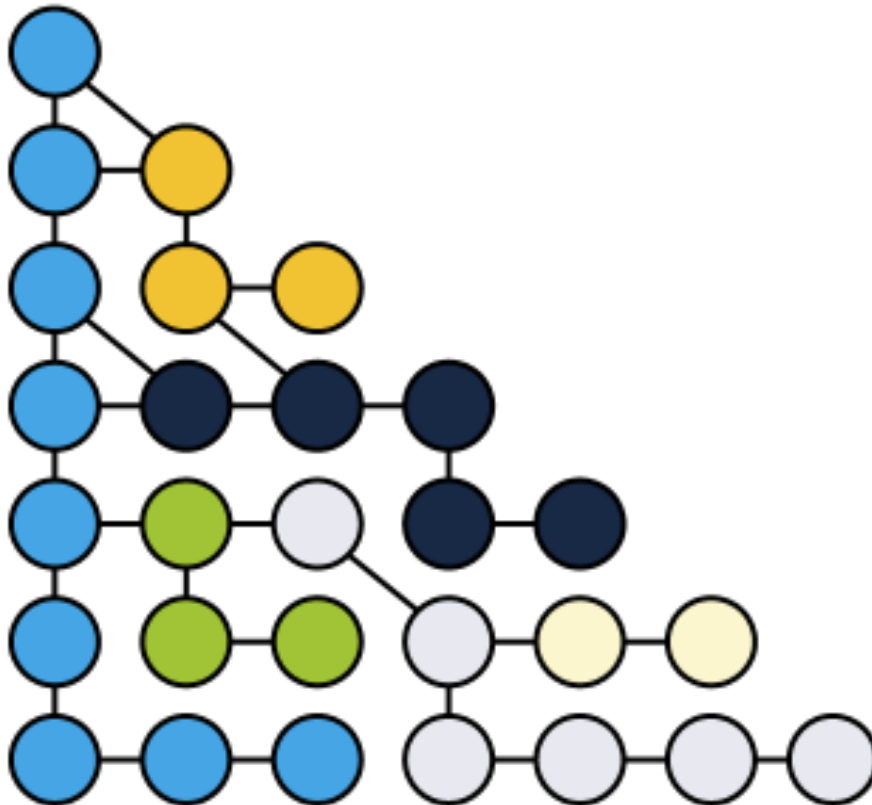


# Thuật toán Depth First Search (1/2)

- ❖ Trong quá trình tìm kiếm DFS tổ chức lưu trữ danh sách các đỉnh theo kiểu **LIFO** - Last In First Out.

# Thuật toán Depth First Search (2/2)

## Level



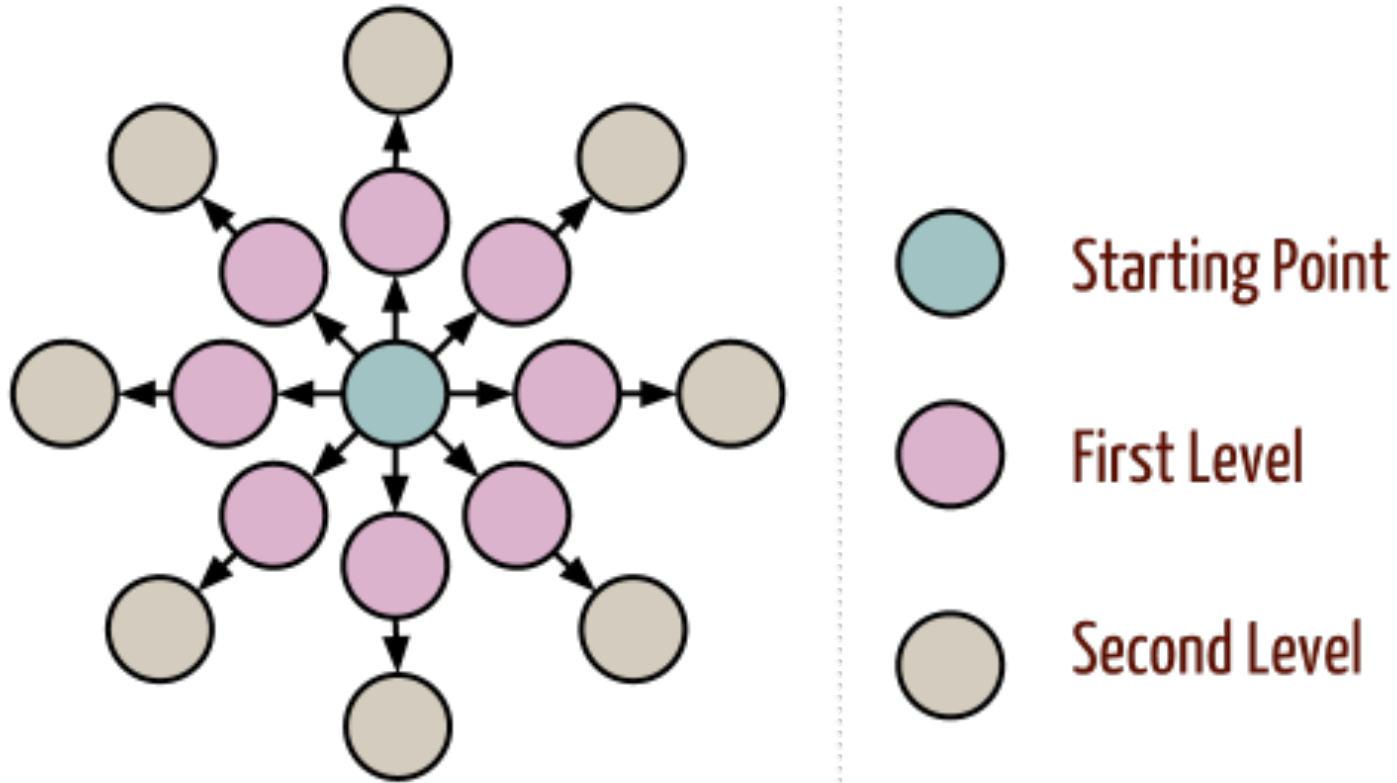


# Thuật toán Breadth First Search (1/3)

- ❖ Trong quá trình tìm kiếm BFS tổ chức lưu trữ danh sách các đỉnh theo kiểu **FIFO** – First In First Out.

# Thuật toán Breadth First Search (2/3)

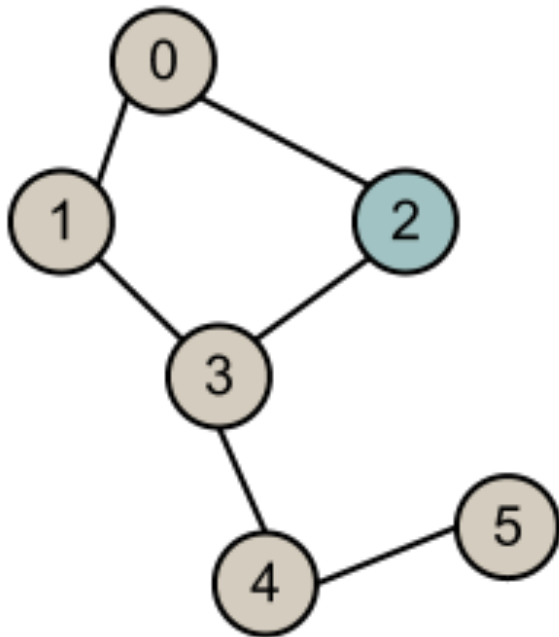
## Breadth First Search



## Wave Approach

# Thuật toán Breadth First Search (3/3)

## Breadth First Search



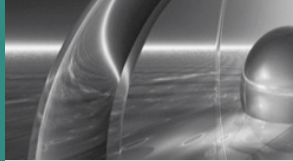
1.  $q = \{\}$
2.  $q = \{2\}$
3.  $q = \{0, 3\}$
4.  $q = \{1\}$
5.  $q = \{4\}$
6.  $q = \{5\}$

Using a queue

## Thảo luận & bài tập (1/5)

❖ Độ phức tạp tính toán của DFS tùy thuộc vào đồ thị được biểu diễn dưới dạng nào, trong trường hợp xấu nhất ta có bảng so sánh sau:

Ma trận kề	Danh sách kề	Danh sách cạnh
$O(n^2)$	$O(n \times m)$	$O(n \times m)$

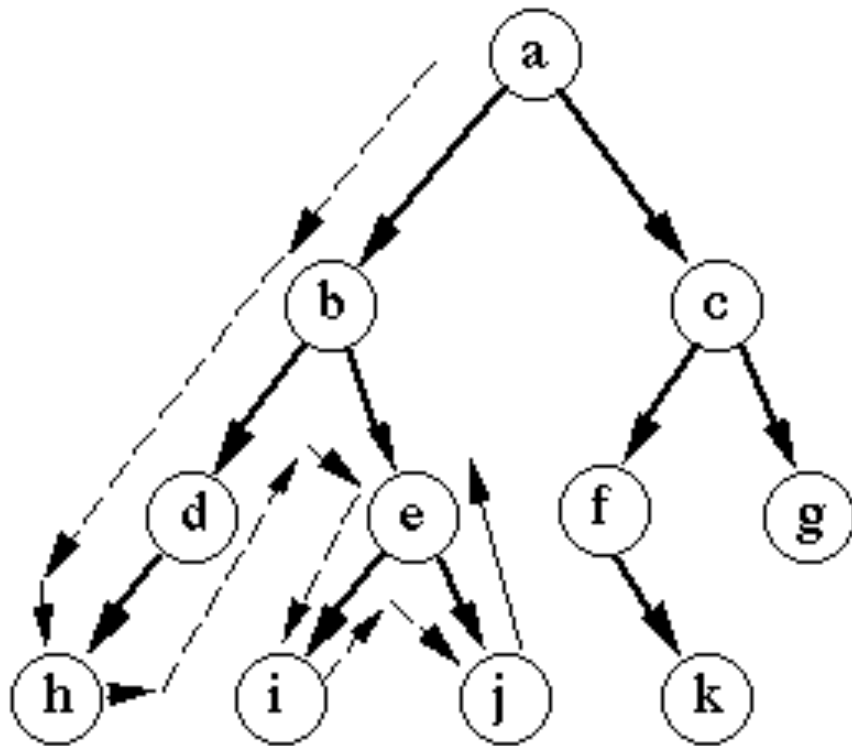


## ❖ Đặc điểm của 2 thuật toán (1/2):

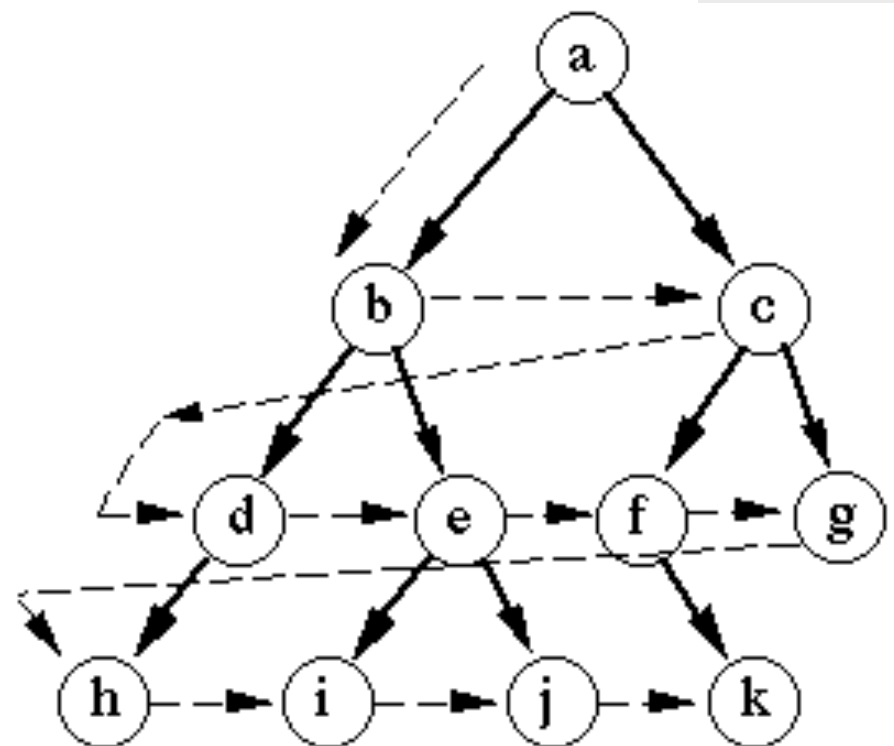
- BFS cho kết quả là đường đi ngắn nhất (tính theo số cung/cạnh).
- Do đó nếu đích “ở gần” đỉnh xuất phát thì BFS **có thể** cho kết quả nhanh hơn DFS.
- Ngược lại nếu đích “xa” đỉnh xuất phát, BFS có thể cho kết quả chậm hơn DFS.
- Đối với đồ thị dạng cây và có độ sâu lớn DFS có thể đi vào “ngõ cụt” do sự bùng nổ tổ hợp.



## ❖ Đặc điểm của 2 thuật toán (2/2):

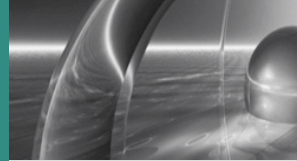


Depth-first search



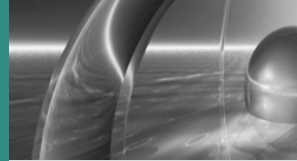
Breadth-first search

# Thảo luận & bài tập (4/5)



Bài toán	Loại đồ thị	DFS hoặc BFS	Bổ sung	Kết quả
Xác định có/không đường đi từ đỉnh $u$ đến đỉnh $v$	Vô hướng	DFS( $u$ ) hoặc DFS( $v$ )	Mỗi khi duyệt xong một đỉnh cần đổi trạng thái của đỉnh đó	?
	Có hướng	DFS( $u$ )		
Tìm đường đi từ đỉnh $u$ đến đỉnh $v$	Vô hướng	DFS( $u$ ) hoặc DFS( $v$ )	Trong quá trình duyệt cần truy vết lại các đỉnh đã đi qua	?
	Có hướng	DFS( $u$ )		

# Thảo luận & bài tập (5/5)



Bài toán	Loại đồ thị	DFS hoặc BFS	Bổ sung	Kết quả
Xác định các thành phần liên thông	Vô hướng	?	?	?
	Có hướng	?	?	?
Duyệt mỗi đỉnh của đồ thị 1 lần	Vô hướng	?	?	?
	Có hướng	?	?	?
Kiểm tra tính liên thông	Vô hướng	?	?	?
Kiểm tra tính liên thông mạnh/yếu	Có hướng	?	?	?
Cài đặt các thuật toán trên	?			