



LÝ THUYẾT ĐỒ THỊ

Graph Theory

Các thuật toán duyệt đồ thị (Graph Searching, Graph Traversal)

- 1. Tìm kiếm theo chiều rộng (BFS)**
- 2. Tìm kiếm theo chiều sâu (DFS)**

Các thuật toán duyệt đồ thị

- Duyệt đồ thị: Graph Searching hoặc Graph Traversal
 - Duyệt qua mỗi đỉnh và mỗi cạnh của đồ thị
- Ứng dụng:
 - Cần để khảo sát các tính chất của đồ thị
 - Là thành phần cơ bản của nhiều thuật toán
- Hai thuật toán duyệt cơ bản:
 - Tìm kiếm theo chiều rộng (Breadth First Search – BFS)
 - Tìm kiếm theo chiều sâu (Depth First Search – DFS)

1. Tìm kiếm theo chiều rộng (Breadth-first Search)

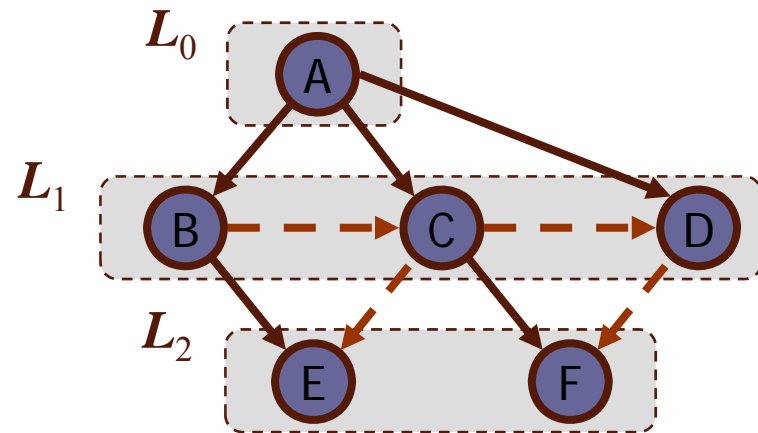
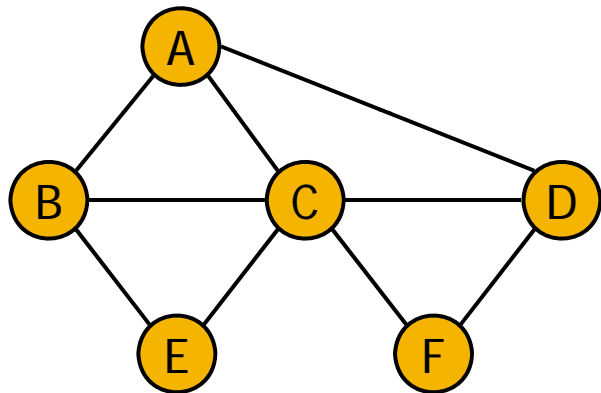
- **Input:** Đồ thị $G = (V, E)$, vô hướng hoặc có hướng.
- **Output:**
 - $d[v]$ = khoảng cách (độ dài của đường đi ngắn nhất) từ s (là đỉnh xuất phát tìm kiếm) đến v , với mọi $v \in V$. $d[v] = \infty$ nếu v không đạt tới được từ s .
 - $\pi[v] = u$ đỉnh đi trước v trong đường đi từ s (là đỉnh xuất phát tìm kiếm) đến v có độ dài $d[v]$.
 - Xây dựng cây BFS với gốc tại s chứa tất cả các đỉnh đạt tới được từ s .

Phân tích BFS

- Việc khởi tạo đòi hỏi $O(|V|)$.
- Vòng lặp duyệt
 - Mỗi đỉnh được nạp vào và loại ra khỏi hàng đợi một lần, mỗi thao tác đòi hỏi thời gian $O(1)$. Như vậy tổng thời gian làm việc với hàng đợi là $O(V)$.
 - Danh sách kề của mỗi đỉnh được duyệt qua đúng một lần. Tổng độ dài của tất cả các danh sách kề là $\Theta(|E|)$.
- Tổng cộng ta có thời gian tính của BFS(s) là $O(|V| + |E|)$, là tuyến tính theo kích thước của danh sách kề biểu diễn đồ thị.

BFS – Loang trên đồ thị

- Thứ tự thăm đỉnh nhờ thực hiện BFS(A)



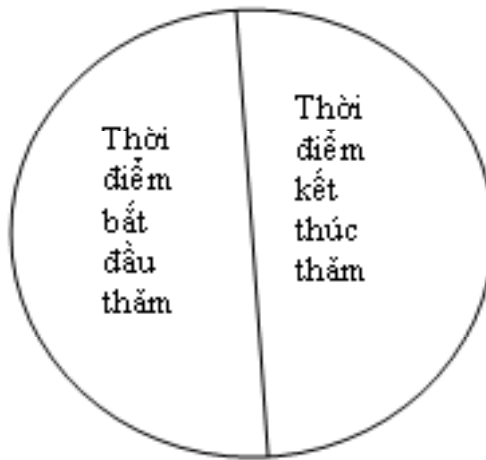
Đầu tiên đến thăm các đỉnh đạt được từ s bởi đường đi qua một cạnh, sau đó là thăm các đỉnh đạt được từ s bởi đường đi qua 2 cạnh... Do đó, nếu đỉnh t được thăm trong BFS(s) thì nó sẽ được thăm theo đường đi ngắn nhất theo số cạnh.

Ứng dụng trực tiếp của BFS

- Sử dụng BFS để kiểm tra tính liên thông của đồ thị vô hướng:
 - Mỗi lần gọi đến BFS ở trong chương trình chính sẽ sinh ra một thành phần liên thông
- Xét sự tồn tại đường đi từ đỉnh s đến đỉnh t :
 - Thực hiện BFS(s).
 - Nếu $\pi[t] = \text{NIL}$ thì không có đường đi, trái lại ta có đường đi
$$t \leftarrow \pi[t] \leftarrow \pi[\pi[t]] \leftarrow \dots \leftarrow s$$
- Chú ý: BFS tìm được đường đi ngắn nhất theo số cạnh.

2. Tìm kiếm theo chiều sâu (Depth-First Search)

- Đánh số các đỉnh theo thời điểm (thứ tự) duyệt đồ thị:
 - Thời điểm bắt đầu thăm
 - Thời điểm kết thúc thăm



Ý tưởng của tìm kiếm theo chiều sâu

- Bắt đầu tìm kiếm từ một đỉnh u nào đó của đồ thị. Sau đó, chọn v là đỉnh chưa được thăm và kề với u , lặp lại quá trình đối với v .
- Ở bước tổng quát, giả sử đang thăm đỉnh v :
 - Nếu trong số các đỉnh kề với v tìm được đỉnh w là chưa được thăm thì ta sẽ thăm đỉnh này (nó sẽ trở thành đã thăm nhưng chưa duyệt xong) và bắt đầu từ nó ta sẽ tiếp tục quá trình tìm kiếm.
 - Nếu như không còn đỉnh nào kề với v là chưa thăm thì ta sẽ nói rằng đỉnh này là đã duyệt xong và quay trở lại tiếp tục tìm kiếm từ đỉnh mà trước đó ta đến được đỉnh v (nếu $v = u$ thì kết thúc tìm kiếm).

Mô tả DFS

- **Input:** Đồ thị $G = (V, E)$ cho bởi danh sách kề
- **Output:**
 - 2 mốc thời gian cho mỗi đỉnh (là các số nguyên trong khoảng 1 và $2|V|$).
 - $d[v] = \text{thời điểm bắt đầu thăm}$ (v chuyển từ trắng sang xám)
 - $f[v] = \text{thời điểm kết thúc thăm}$ (v chuyển từ xám sang đen)
 - $\pi[v]$: đỉnh đi trước v – tức là đỉnh mà từ đó ta đến thăm v .
- Sử dụng cách biến màu như đã mô tả.

Phân tích DFS

- Vòng lặp trên các dòng 1-2 và 5-7 đòi hỏi thời gian $\Theta(|V|)$, chưa tính thời gian thực hiện lệnh DFS(v).
- DFS(v) thực hiện đối với mỗi đỉnh trắng $v \in V$ và ngay sau khi được thăm nó được tô màu xám. Các dòng 3-6 của DFS(v) sẽ thực hiện $|\text{Adj}[v]|$ lần. Vậy thời gian tổng cộng của DFS(v) là $\sum_{v \in V} |\text{Adj}[v]| = \Theta(|E|)$
- Do đó thời gian của DFS là $\Theta(|V| + |E|)$.
- Thuật toán trên đồ thị có đánh giá thời gian như trên gọi là thuật toán thời gian tuyến tính

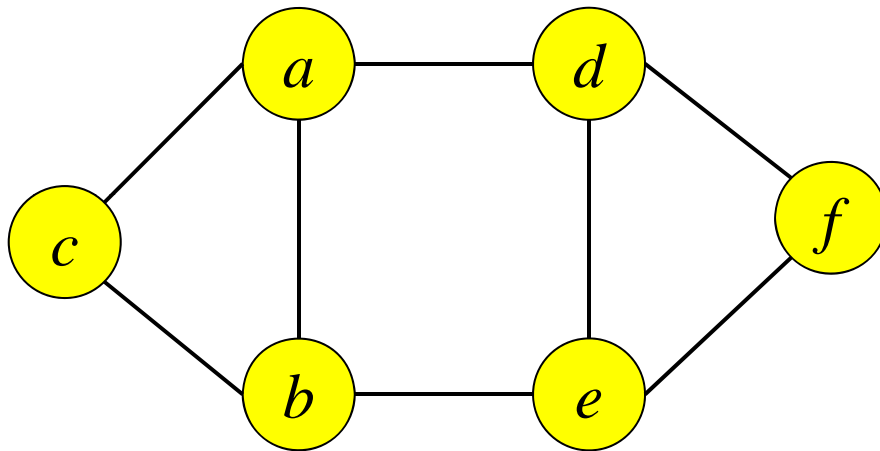
CÁC ỨNG DỤNG CỦA DFS

- Tính liên thông của đồ thị
- Tìm đường đi từ s đến t
- Phát hiện chu trình
- Kiểm tra tính liên thông mạnh
- Định hướng đồ thị

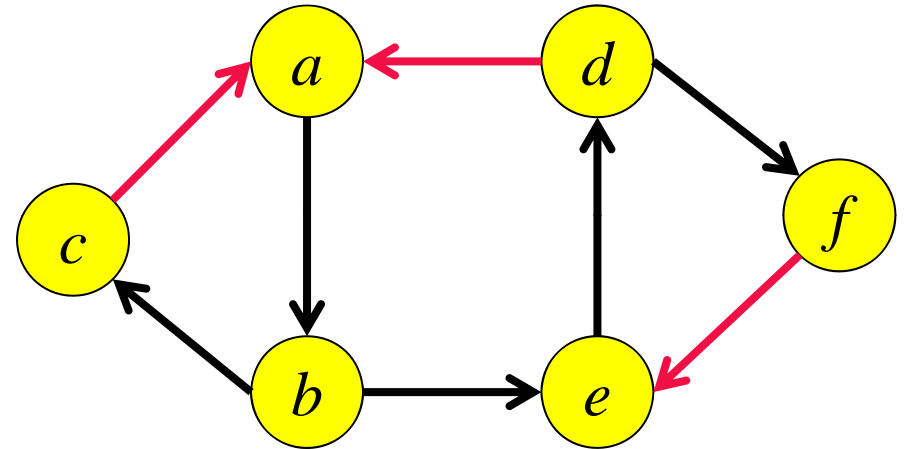
Định hướng đồ thị

- **Bài toán:** Cho đồ thị vô hướng liên thông $G = (V, E)$. Hãy tìm cách định hướng các cạnh của nó để thu được đồ thị có hướng liên thông mạnh hoặc trả lời G là không định hướng được.
- **Thuật toán định hướng δ :** Trong quá trình thực hiện DFS(G) định hướng các cạnh của cây DFS theo chiều từ tổ tiên đến con cháu, các cạnh ngược theo hướng từ con cháu đến tổ tiên. Ký hiệu đồ thị thu được là $G(\delta)$
- **Bổ đề.** G là định hướng được khi và chỉ khi $G(\delta)$ là liên thông mạnh.

Ví dụ: Định hướng đồ thị



Đồ thị G



Đồ thị $G(\delta)$