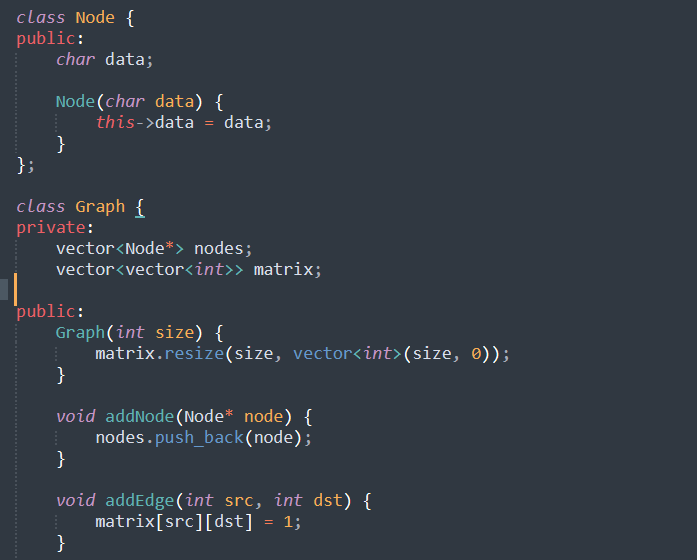
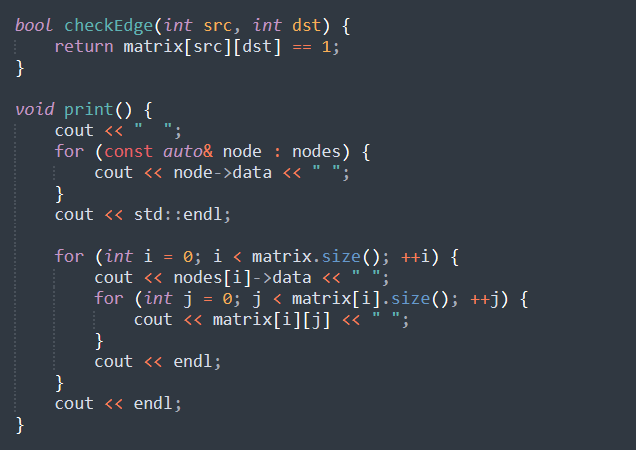


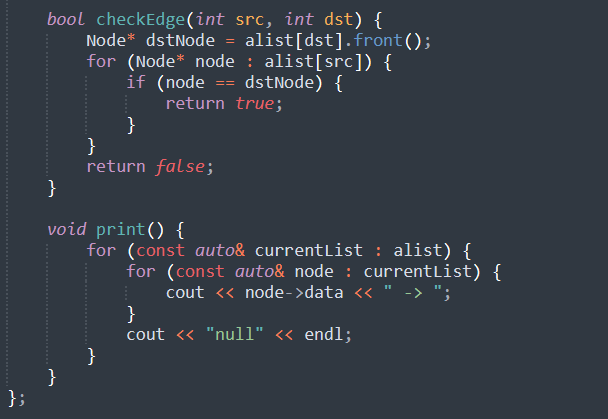
1. Adjacency Matrix



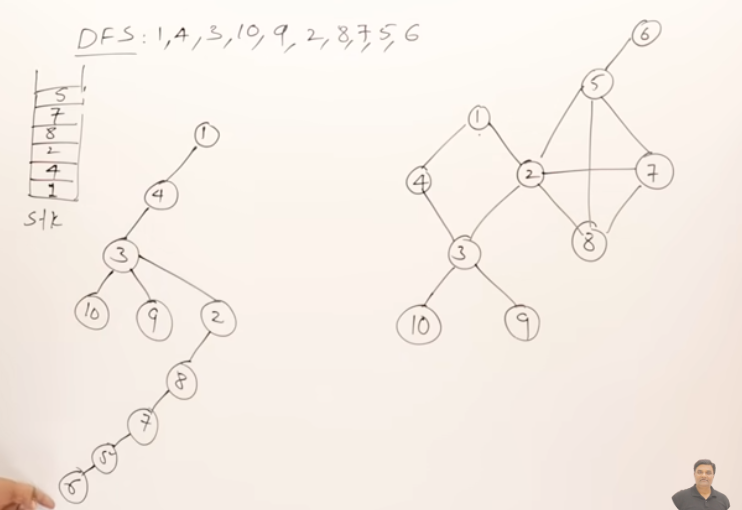


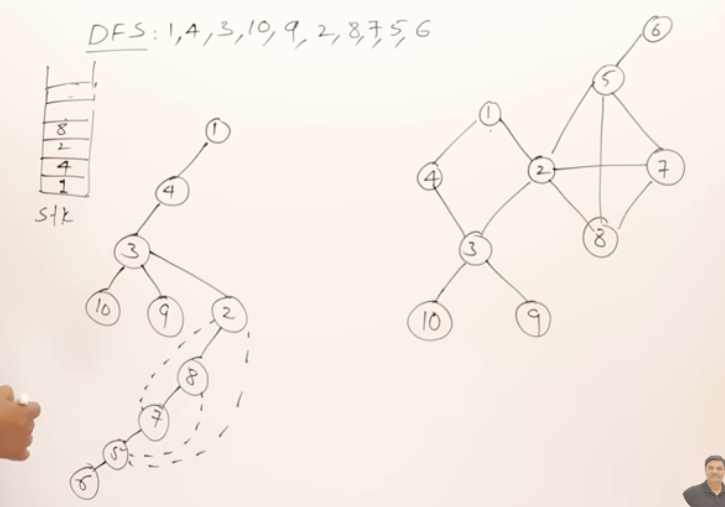
1. Adjacency List

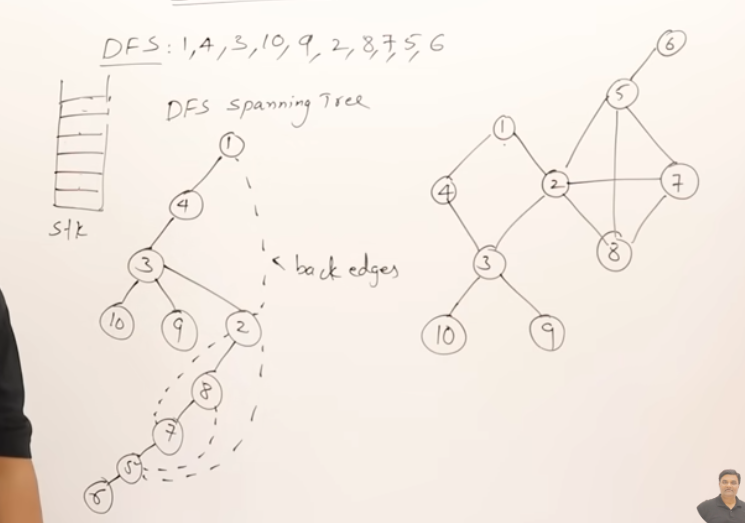




1. Depth First Search

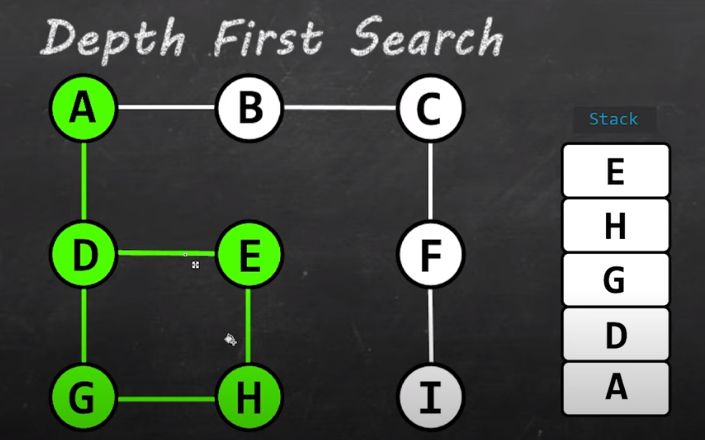


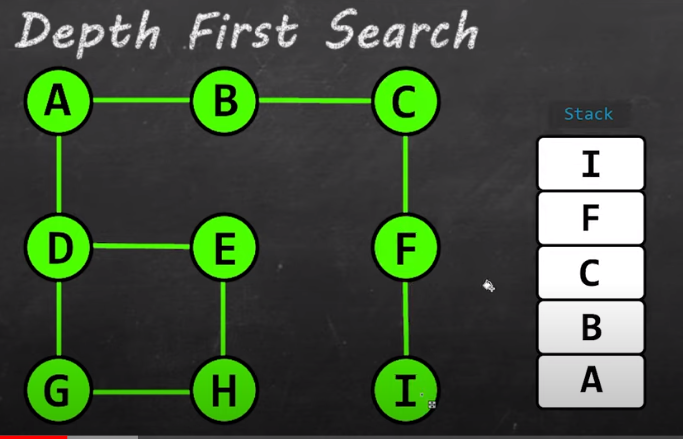




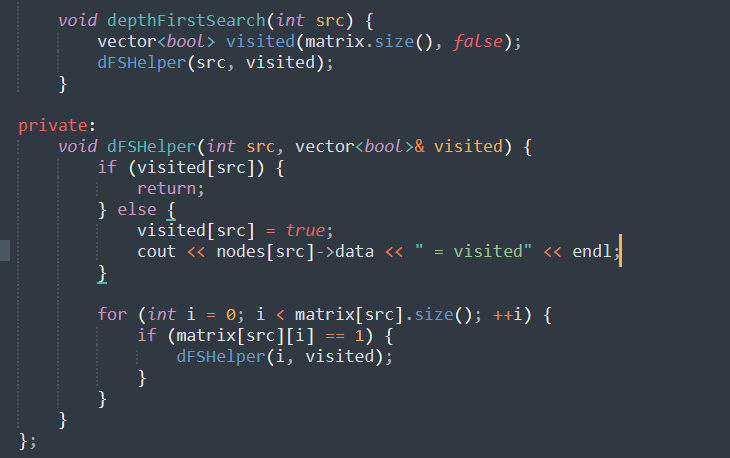
**Chú ý:**

* Duyệt sâu vào từng nút kề với nút đầu tiên được xét, mỗi khi đang duyệt sau vào 1 nút nào thì phải thêm nút đang duyệt đó vào stack để đánh dấu địa điểm để còn biết quay lại khi đi sâu vào điểm đó mà gặp ngõ cụt. Mỗi khi quay lại điểm đó thì xóa nút đó đi và nếu vẫn còn nút kề với nút đó (đường đi mới) thì lại thêm nút đó vào stack rồi đi sâu vào tiếp nút kề mới đó của nó. Lặp lại quá trình cho đến khi stack rỗng và tất cả các nút đã được thăm hết là thỏa mãn.

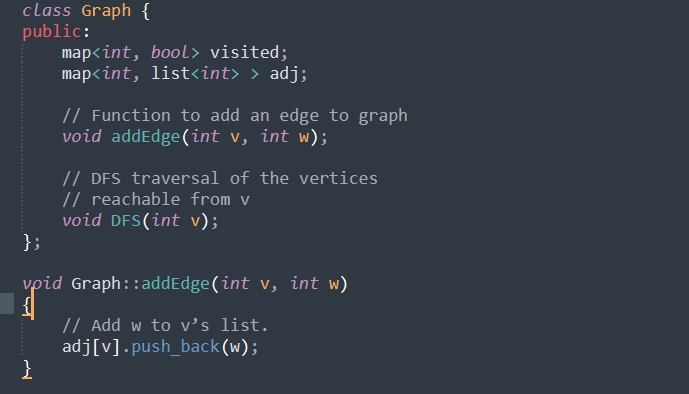


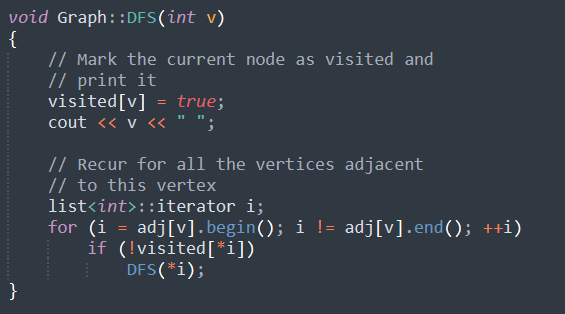


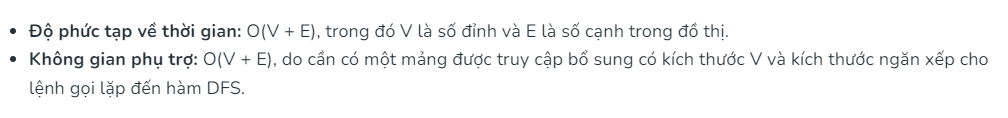
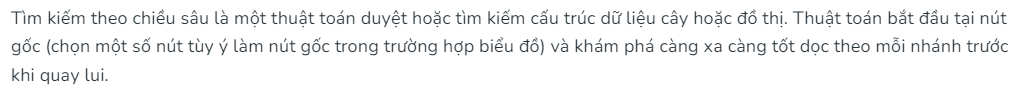
**Biểu diễn bằng ma trận kề**

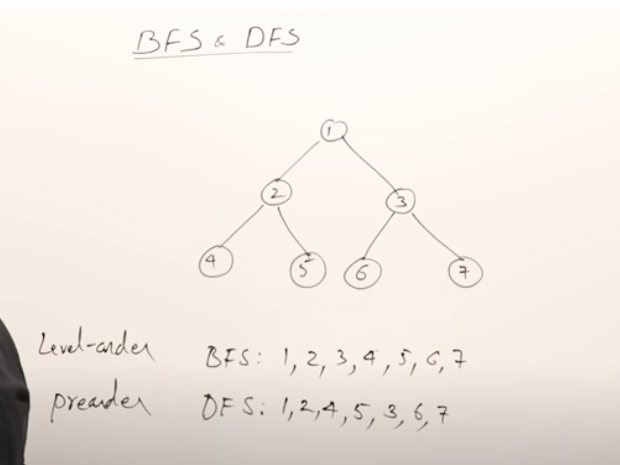


**Biểu diễn bằng danh sách kề**

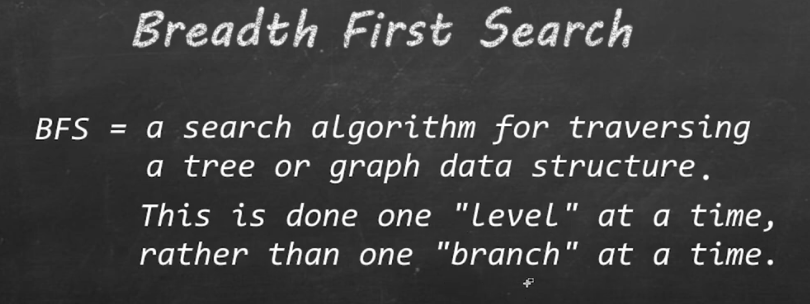
****

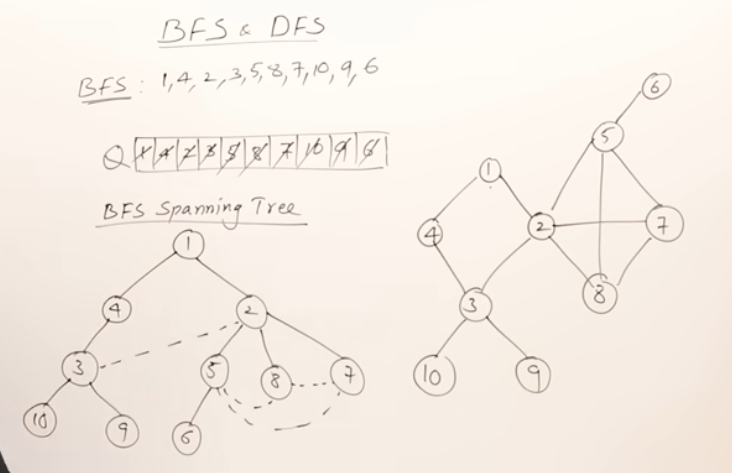
****





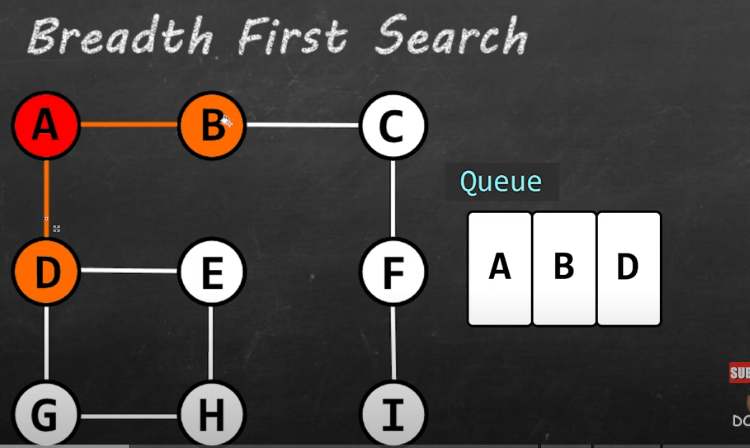
1. Breadth First Search

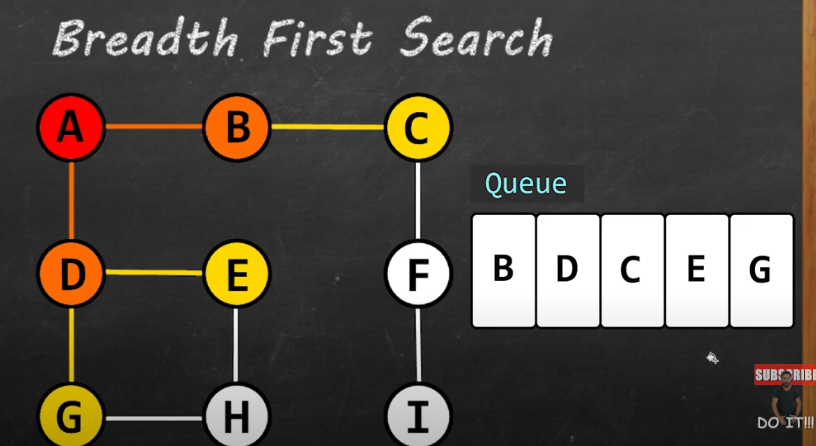


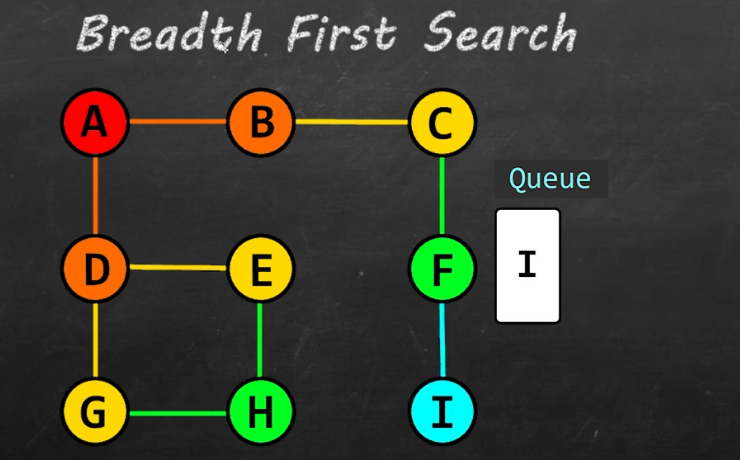


Chú ý: - Bạn có thể chọn bất kì đỉnh nào để ghé thăm đầu tiên sau đấy đẩy nó vào hàng đợi (queue)

* Tiếp theo bạn xóa 1 ra khỏi hàng đợi và hãy thăm các đỉnh liền kề của nó rồi lại đẩy vào queue
* Nếu thăm đến một đỉnh đã được thăm bởi nút khác trước đó thì hãy nối bằng nét đứt đến nút đó.



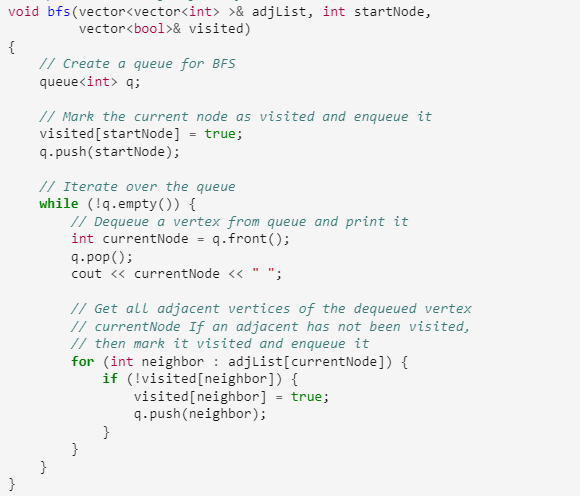


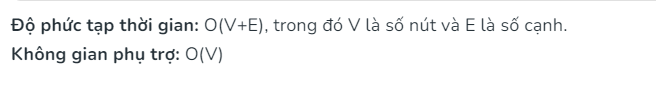


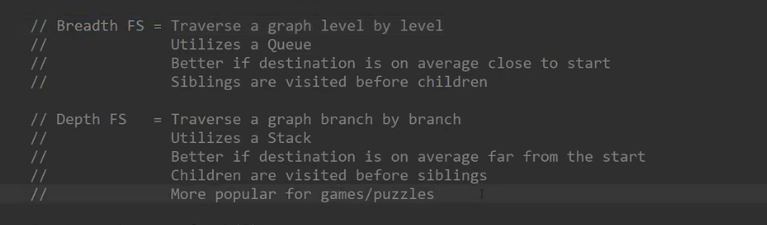
**Duyệt bằng ma trận kề**



**Bằng danh sách kề**

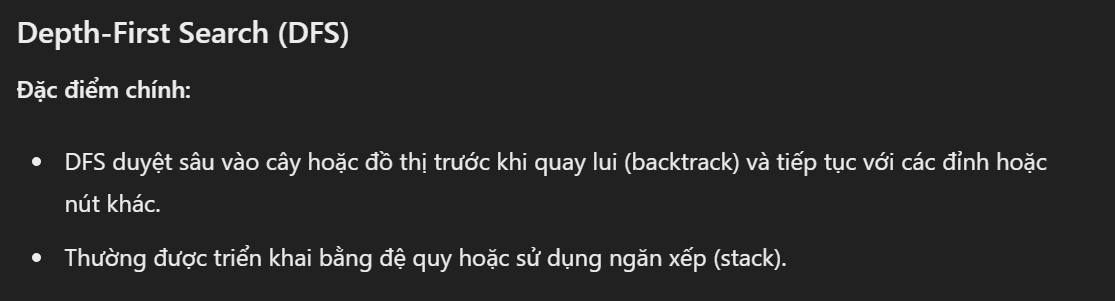
****

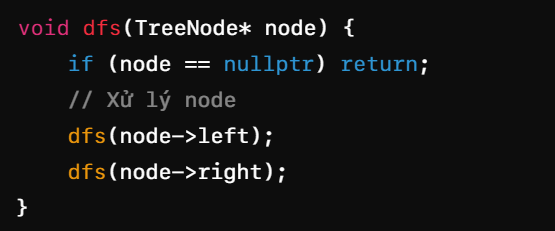
****

****

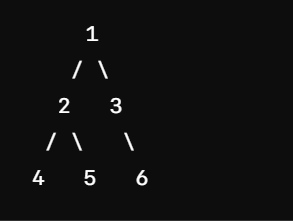
**Áp dụng DFS, BFS khi code algorithms**

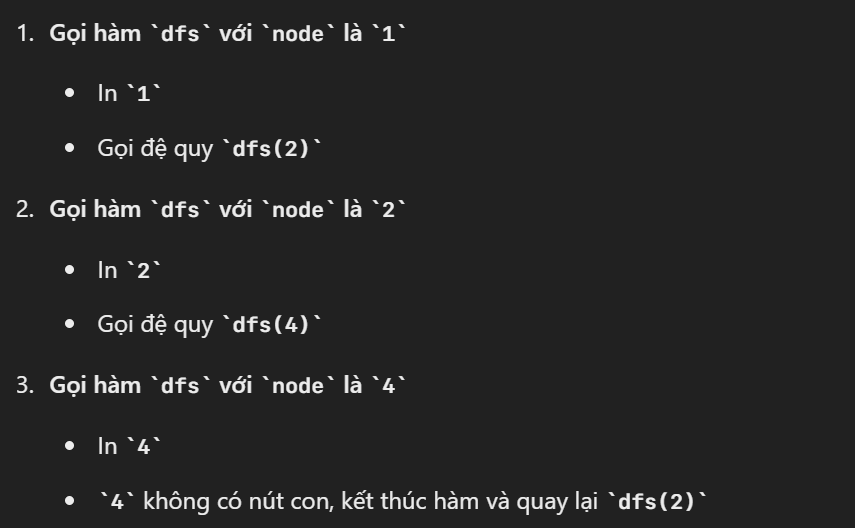
**1, DFS**

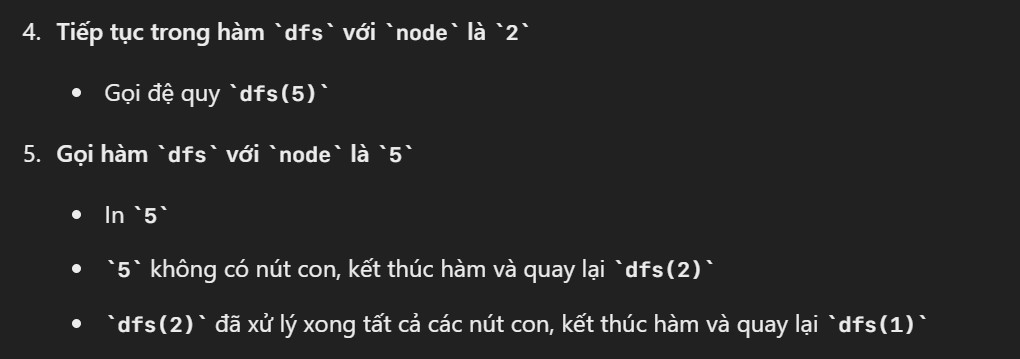
1, Dùng đệ quy

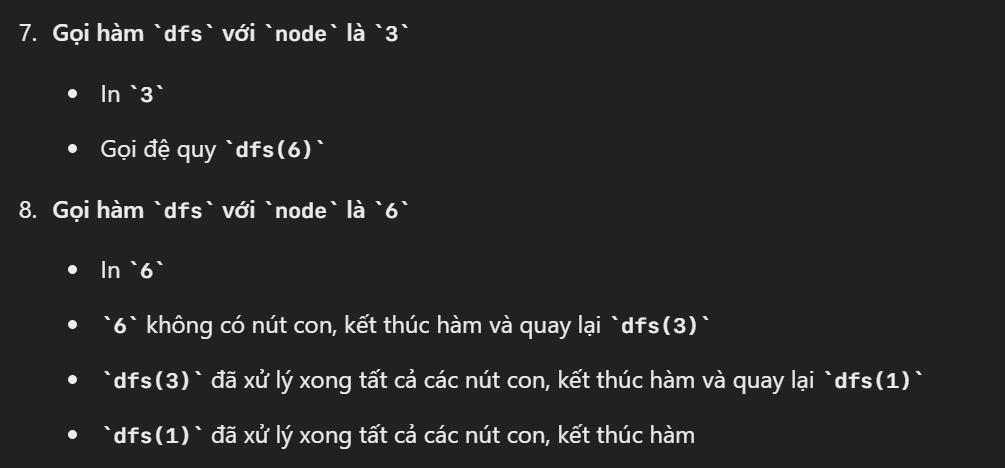


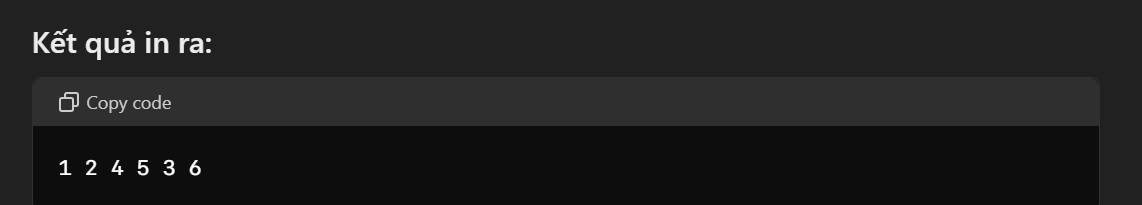
Ví dụ:







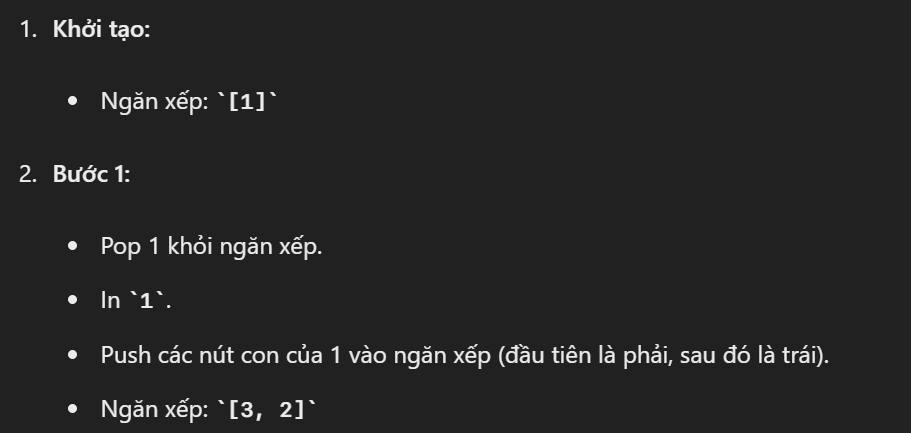


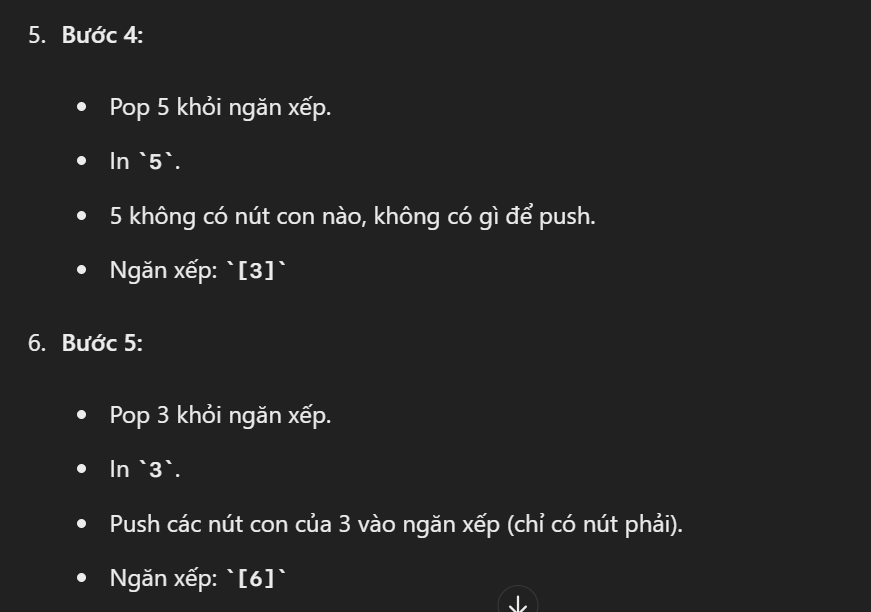
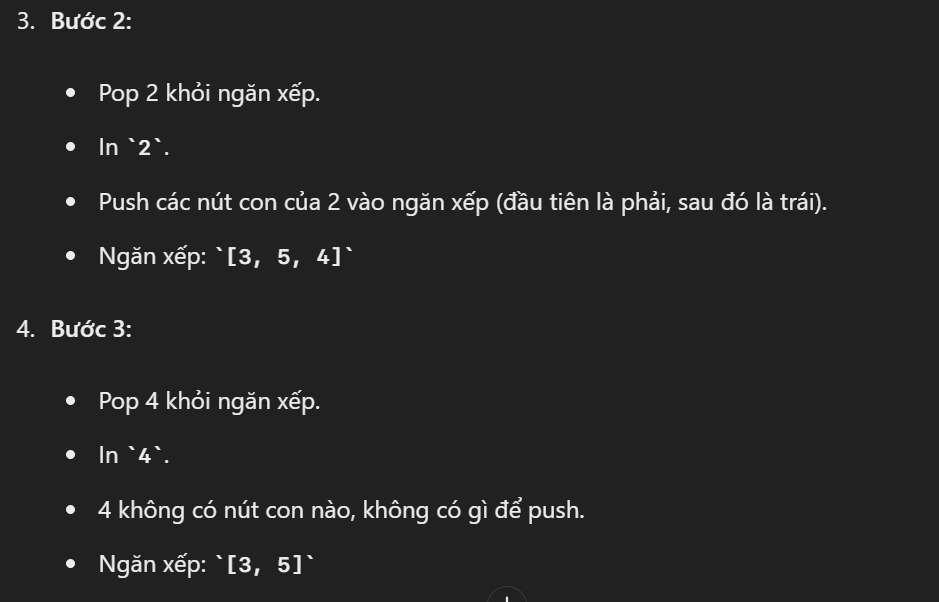


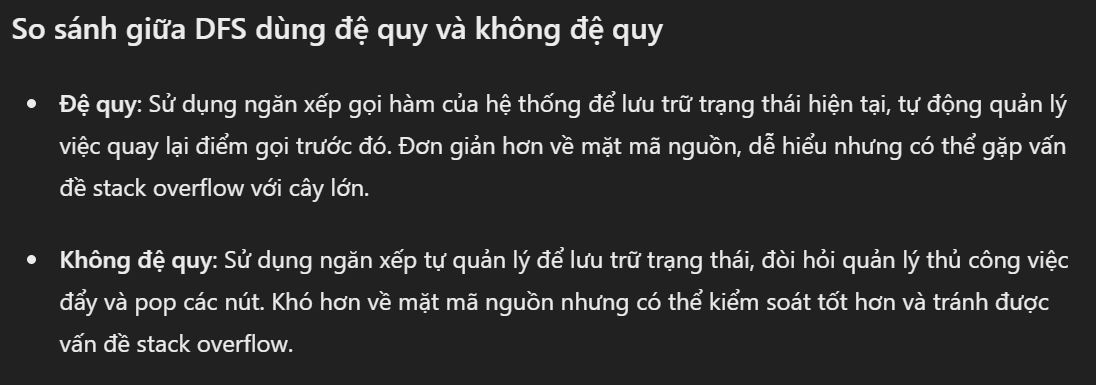
2, Dùng Stack



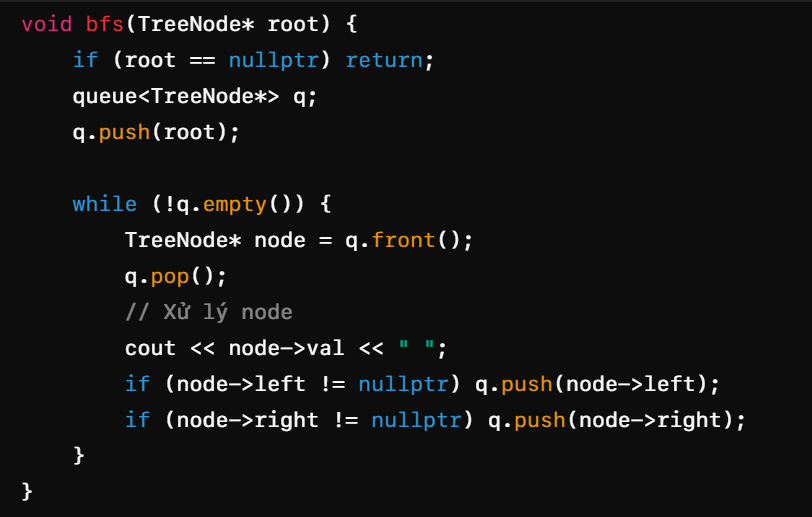
Ví dụ:







**2, BFS : dùng queue**

****

**Ví dụ:**

