**TRƯỜNG ĐẠI HỌC BÁCH KHOA HÀ NỘI   
VIỆN CÔNG NGHỆ THÔNG TIN VÀ TRUYỀN THÔNG**

────────\*--───────



**CÁC THUẬT TOÁN TÌM ĐƯỜNG ĐI TRONG ĐỒ THỊ**

**TRÍ TUỆ NHÂN TẠO**

**Giáo viên hướng dẫn: Nguyễn Ngọc Bích**

**Sinh viên thực hiện: Đặng Văn Đạt - 20158081**

**Hà Nội, năm 2020**

# 

**MỤC LỤC**

[**LỜI NÓI ĐẦU 3**](#_Toc57232266)

[**1 Phân tích yêu cầu đề tài 5**](#_Toc57232267)

[**1.1 Mục đích 5**](#_Toc57232268)

[**2 Lý thuyết 5**](#_Toc57232269)

[**2.1 Biểu diễn đồ thị 5**](#_Toc57232270)

[**2.2 Thuật toán tìm đường đi 7**](#_Toc57232271)

[**2.2.1 Tìm kiếm sâu: 7**](#_Toc57232272)

[**3 Cài đặt trong ngôn ngữ Java 11**](#_Toc57232273)

[**3.1 Code trong Java. 11**](#_Toc57232274)

[**4 Tổng kết 24**](#_Toc57232275)

# LỜI NÓI ĐẦU

Ngày nay, Trí tuệ nhân tạo hay trí thông minh nhân tạo (Artificial intelligence – viết tắt là AI) là một ngành thuộc lĩnh vực khoa học máy tính (Computer science). Là trí tuệ do con người lập trình tạo nên với mục tiêu giúp máy tính có thể tự động hóa các hành vi thông minh như con người.

Trí tuệ nhân tạo khác với việc lập trình logic trong các ngôn ngữ lập trình là ở việc ứng dụng các hệ thống học máy (machine learning) để mô phỏng trí tuệ của con người trong các xử lý mà con người làm tốt hơn máy tính.

Cụ thể, trí tuệ nhân tạo giúp máy tính có được những trí tuệ của con người như: biết suy nghĩ và lập luận để giải quyết vấn đề, biết giao tiếp do hiểu ngôn ngữ, tiếng nói, biết học và tự thích nghi, …

Bên cạnh những ứng dụng hiện hành, trí tuệ nhân tạo đã từng giải quyết một bài toán vô cùng cổ điển trong tin học, đó là bài toán tìm đường đi của đồ thị. Dưới sự hướng dẫn của cô Nguyễn Ngọc Bích, hôm nay em xin trình bài tập lớn Các phương pháp tìm đường đi trong đồ thị.

# Phân tích yêu cầu đề tài

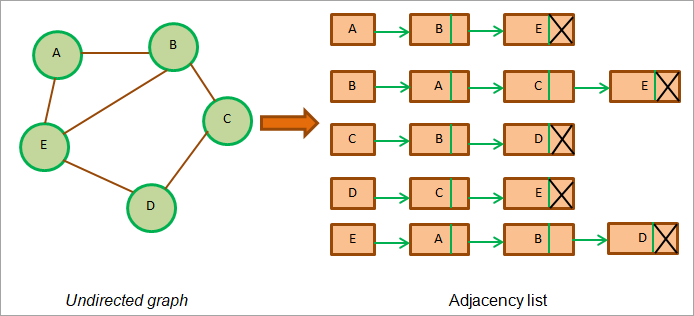
## Mục đích

* Tìm hiểu lý thuyết các phương pháp tìm đường đi trong đồ thị bao gồm:
  + Tìm kiếm sâu (Depth First Search –DFS)
* Áp dụng lý thuyết nêu trên và thực hành cài đặt với 2 loại cấu trúc giữ liệu:
  + Danh sách móc nối (hai chiều)

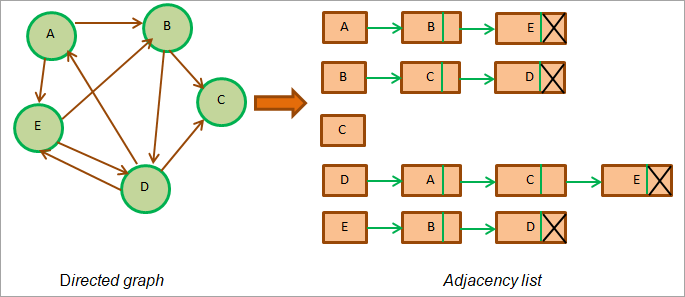
# Lý thuyết

## Biểu diễn đồ thị

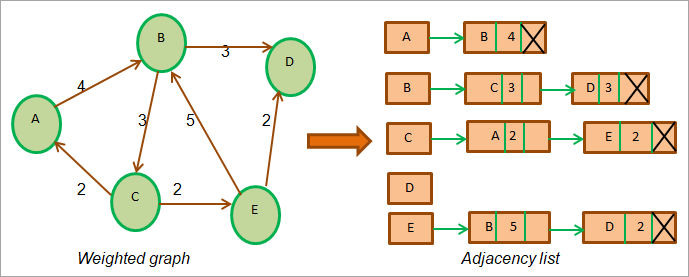
* Trong việc biểu điễn đồ thị, ta sẽ dùng Danh sách liền kề cho cả 3 loại đồ thị
  + Vô hướng có trọng số.
  + Vô hướng không trọng số.
  + Có hướng có trọng số.
  + Có hướng không trọng số.
* Biểu diễn liên kết này được gọi là danh sách kề. Một danh sách kề không là gì ngoài một danh sách được liên kết và mỗi nút trong danh sách đại diện cho một đỉnh.
* Sự hiện diện của một cạnh giữa hai đỉnh được biểu thị bằng một con trỏ từ đỉnh thứ nhất đến đỉnh thứ hai. Danh sách kề này được duy trì cho mỗi đỉnh trong biểu đồ.
* Khi chúng tôi đã duyệt qua tất cả các nút lân cận cho một nút cụ thể, chúng tôi lưu trữ NULL trong trường con trỏ tiếp theo của nút cuối cùng của danh sách kề.
* Bây giờ chúng ta sẽ sử dụng các đồ thị trên mà chúng ta đã sử dụng để biểu diễn ma trận kề để chứng minh danh sách kề.
* Ví dụ:



Hình 1. Đồ Thị vô hướng không trọng số



Hình 2. Đồ Thị vô có hướng không trọng số



Hình 3. Đồ Thị có hướng có trọng số

## Thuật toán tìm đường đi

### Tìm kiếm sâu:

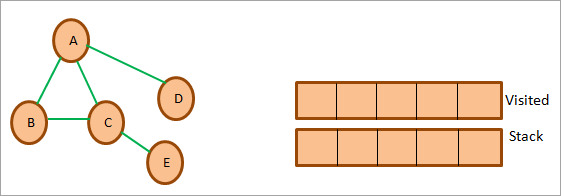
#### Giới thiệu:

* Tìm kiếm theo độ sâu trước tiên (DFS) là một kỹ thuật được sử dụng để duyệt qua một cây hoặc một đồ thị. Kỹ thuật DFS bắt đầu với một nút gốc và sau đó đi qua các nút lân cận của nút gốc bằng cách đi sâu hơn vào biểu đồ. Trong kỹ thuật DFS, các nút được duyệt theo chiều sâu cho đến khi không còn nút nào để khám phá.
* Khi chúng ta đến nút lá (không còn nút con nữa), DFS sẽ lùi lại và bắt đầu với các nút khác và thực hiện truyền tải theo cách tương tự. Kỹ thuật DFS sử dụng cấu trúc dữ liệu ngăn xếp để lưu trữ các nút đang được duyệt.

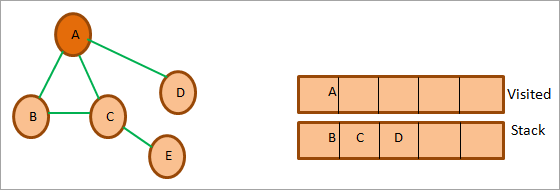
#### Cách làm

* Bước 1: Bắt đầu với nút gốc và chèn nó vào ngăn xếp
* Bước 2: Mở mục từ ngăn xếp và chèn vào danh sách 'đã ghé thăm
* Bước 3: Đối với nút được đánh dấu là ‘đã thăm’ (hoặc trong danh sách đã truy cập), hãy thêm các nút liền kề của nút này chưa được đánh dấu là đã thăm, vào ngăn xếp.
* Bước 4: Lặp lại bước 2 và 3 cho đến khi ngăn xếp trống.

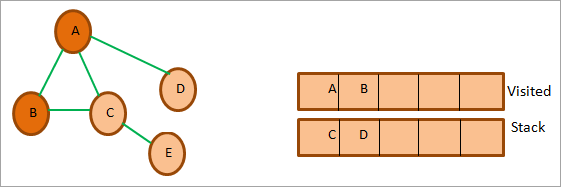
#### Ví dụ:



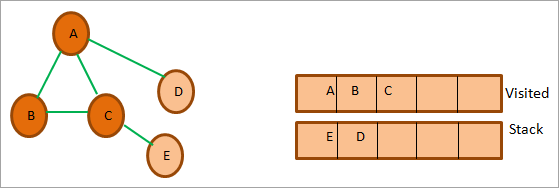
Chúng ta sẽ bắt đầu với A để bắt đầu, đánh dấu nó là đã thăm và thêm nó vào danh sách đã ghé thăm.Sau đó, chúng ta sẽ xem xét tất cả các nút lân cận của A và đẩy các nút này lên ngăn xếp như hình dưới đây.



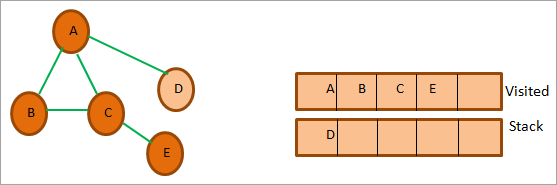
Tiếp theo, chúng tôi bật một nút từ ngăn xếp, tức là B và đánh dấu nó là đã thăm. Sau đó, chúng tôi thêm nó vào danh sách ‘đã ghé thăm’.



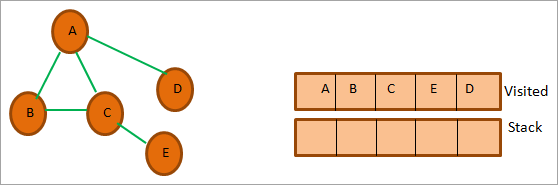
Bây giờ chúng ta xem xét các nút lân cận của B là A và C. Trong số này A đã được thăm. Vì vậy, chúng tôi bỏ qua nó. Tiếp theo, chúng tôi bật C từ ngăn xếp. Đánh dấu C là đã thăm. Nút liền kề của C tức là E được thêm vào ngăn xếp.



Tiếp theo, chúng tôi bật nút tiếp theo E từ ngăn xếp và đánh dấu nó là đã thăm. Nút liền kề của nút E là nút C đã được truy cập. Vì vậy, chúng tôi bỏ qua nó.



Bây giờ chỉ có nút D còn lại trong ngăn xếp. Vì vậy, chúng tôi đánh dấu nó là đã thăm. Nút liền kề của nó là A đã được truy cập. Vì vậy, chúng tôi không thêm nó vào ngăn xếp.



Tại thời điểm này, ngăn xếp trống. Điều này có nghĩa là chúng tôi đã hoàn thành việc duyệt theo độ sâu đầu tiên cho biểu đồ đã cho.

Danh sách được truy cập cung cấp trình tự duyệt cuối cùng bằng cách sử dụng kỹ thuật theo chiều sâu. Dãy DFS cuối cùng cho đồ thị trên là A-> B-> C-> E-> D.

# Cài đặt trong ngôn ngữ Java

## Code trong Java.

1. **Cài đặt Danh sách móc nối 2 chiều cho Stack và Queue**

* class **Node**:





* class **MixLinkedList**:









1. **Cài đặt đồ thị**

* Class **Edge:**

****

* Class **Graph:**

****

* Test Graph

Diagram, shape

Description automatically generated

* Code



* Kết quả:



1. **Cài đặt thuật toán Tìm Kiếm Sâu (Depth-first search - DFS)**

* Abstract class **SearchAbstract**(là lớp cha của tất cả các thuật toán tìm kiếm)

****

* Class **DFSearch:**

****

****



* Test trên cây đã cho:

Diagram, shape

Description automatically generated



* Kết quả:



# Tổng kết

* Qua Thời gian tìm hiểu về các thuật toán vế tìm đường đi trên đồ thị dưới sự hướng dẫn của cô, em đã học được thêm những kiến thức nền tảng đối với bộ môn Trí Tuệ nhân tạo cũng như nắm chắc kiến thức đã học trong bộ môn Cấu trúc dữ liệu và giải thuật thông qua việc áp cấu trúc Mảng và Danh sách móc nối vào cài đặt thuật toán.
* Vì kinh nghiệm thực tế chưa nhiều nên chương trình không tránh khỏi những sai sót, rất mong sự góp ý của cô.