# Thực hành môn Truy hồi thông tin và tìm kiếm web

## Buổi 3. Xử lý câu truy vấn

### Phần I. Đọc hướng dẫn về câu truy vấn trong thư viện Whoosh

### 1.1. Nhắc lại cách tạo đối tượng truy vấn

```
from whoosh.qparser import QueryParser
parser = QueryParser(<trường tìm kiếm>, <đối tượng lược đồ>)
<đối tượng truy vấn> = parser.parse(<xâu truy vấn>)
```

### 1.2. Các phép toán lôgic

- Ngầm định các từ trong câu truy vấn được kết nối với nhau bằng phép `AND`. Hai câu truy vấn sau đây là tương đương:

  ```
  physically based rendering
  physically AND based AND rendering
  ```

- Tìm những tài liệu chứa `render`, đồng thời chứa `shading` hoặc `modeling`:

  ```
  render AND (shading OR modeling)
  ```

- Tìm những tài liệu chứa `render` nhưng không chứa `modeling`:

  ```
  render NOT modeling
  ```

- Tìm những tài liệu chứa `alpha` nhưng không chứa `beta`, không chứa `gamma`:

  ```
  alpha NOT (beta OR gamma)
  ```

### 1.3. Từ truy vấn có chứa ký tự đại diện

Dấu ? biểu diễn một ký tự, còn dấu * biểu diễn một dãy ký tự tùy ý (gồm 0, 1 hoặc nhiều ký tự).

### Phần II. Đọc hiểu, chạy thử và cải tiến chương trình tìm kiếm cho sẵn

### 2.1. Các file và thư mục cho sẵn

**Thư mục CS-Docs:**

- Chứa 5 file tài liệu tiếng Anh về các chủ đề trong công nghệ thông tin: artificial intelligence, machine learning, data mining, software engineering, operating system.
- Mỗi file gồm dòng đầu tiên là tiêu đề và phần còn lại là nội dung.

**File index.py:**

- Chương trình Python để xây dựng chỉ mục.
- Lược đồ của chỉ mục gồm các trường title, path và content.

**File search.py:**

- Chương trình Python để tìm kiếm.
- Người dùng gõ vào câu truy vấn, chương trình hiện ra các kết quả, mỗi kết quả gồm số thứ tự, tên file, tiêu đề và điểm số của tài liệu.
- Người dùng gõ tiếp số thứ tự kết quả để xem nội dung tài liệu.

### 2.2. Chạy chương trình

Chạy chương trình **index.py** trước, sau đó mới chạy chương trình **search.py**. Để chạy một chương trình:

- Mở dấu nhắc Anaconda;
- Gõ lệnh **cd** đổi thư mục hiện hành sang thư mục chứa chương trình;
- Gõ lệnh **python <tên file chương trình>** để chạy chương trình.

### 2.3. Thử nghiệm các kiểu câu truy vấn

Chạy chương trình **search.py** để thử nghiệm:

- Câu truy vấn có chứa các phép toán lôgic AND, OR và NOT;
- Câu truy vấn chứa ký tự đại diện (dấu ? và dấu *).

Chú ý kiểm tra các kết quả trả về có đúng hay không.

### 2.4. Xây dựng giao diện tìm kiếm thân thiện người dùng

Vì người dùng có thể không hiểu các phép toán lôgic, hãy cải tiến chương trình **search.py** để cung cấp giao diện tìm kiếm thân thiện người dùng hơn.

Trước tiên, yêu cầu người dùng nhập vào các từ phải có mặt trong tài liệu, như **learning data**. Sau đó, yêu cầu người dùng nhập tiếp các từ mà chỉ cần một trong các từ đó có mặt trong tài liệu là được, như **text image**. Cuối cùng, yêu cầu người dùng nhập vào những từ không được phép xuất hiện trong tài liệu, như **programming software**. Sau khi đã có các từ truy vấn từ người dùng, hãy viết code để tạo ra câu truy vấn đầy đủ như bên dưới:

**learning data (text OR image) NOT programming NOT software**