

# Tìm kiếm thông tin

## Chương 1. Giải pháp Boolean


*Soạn bởi: TS. Nguyễn Bá Ngọc*

2021

# Nội dung

1. Vấn đề tìm kiếm thông tin
2. Mô hình Boolean
3. Giải thuật đơn giản đánh chỉ mục trong RAM
4. Xử lý truy vấn Boolean
5. Mô hình Boolean mở rộng và xếp hạng

# Nội dung

- 
1. Vấn đề tìm kiếm thông tin
  2. Mô hình Boolean
  3. Giải thuật đơn giản đánh chỉ mục trong RAM
  4. Xử lý truy vấn Boolean
  5. Mô hình Boolean mở rộng và xếp hạng

# Khái niệm tìm kiếm thông tin

Tìm kiếm thông tin (Information retrieval, IR)

*là trích xuất các tài nguyên thông tin (thường là văn bản) có tính chất phi cấu trúc từ trong một nguồn tài nguyên thông tin lớn (thường được lưu trên máy tính), nhằm đáp ứng các nhu cầu thông tin.*

Một số tình huống tìm kiếm thông tin điển hình ngày nay:

- Tìm kiếm thông tin Web - phổ biến nhất;
- Tìm kiếm email;
- Tìm kiếm tài liệu điện tử trên máy tính cá nhân;
- Tra cứu thông tin pháp lý;
- V.V..

# Hệ thống TKTT và CSDL: Dữ liệu có cấu trúc và dữ liệu phi cấu trúc

- Dữ liệu có cấu trúc, tiêu biểu như CSDL quan hệ:
  - Một CSDL quan hệ bao gồm các bảng, một bảng bao gồm nhiều cột được định kiểu, dữ liệu trong bảng được lưu theo dòng, ..
  - Ví dụ một bảng:

| <b>Nhân viên</b> | <b>Quản lý</b> | <b>Lương</b> |
|------------------|----------------|--------------|
| Minh             | Huệ            | 50,000,000   |
| Liên             | Hương          | 60,000,000   |
| Hải              | Huệ            | 70,000,000   |

- Truy xuất dữ liệu từ CSDL quan hệ được thực hiện bằng ngôn ngữ SQL (là một ngôn ngữ hình thức)
  - Ví dụ tìm nhân viên có Lương > 55,000,000 và người quản lý có tên là Huệ:  
`SELECT * FROM NhanVien WHERE luong > 55000000 AND quanly = "Huệ"`

# Hệ thống TKTT và CSDL: Dữ liệu phi cấu trúc

- Dữ liệu phi cấu trúc: Diễn hình là dữ liệu văn bản, không được tổ chức theo một cấu trúc chặt chẽ.
- Trong không gian số tồn tại một lượng rất lớn thông tin phục vụ cuộc sống con người, vì vậy rất hữu ích và cũng có cấu trúc rất linh hoạt, có thể được coi là phi cấu trúc đối với xử lý bằng máy tính.
- Hệ thống TKTT hướng tới xử lý và cho phép truy xuất thông tin một cách hiệu quả từ một nguồn thông tin có tính phi cấu trúc:
  - Thường cho phép truy xuất bằng từ khóa
  - Ví dụ tìm tài liệu về *công nghệ thông tin*

# Dữ liệu bán cấu trúc

- Dữ liệu bán cấu trúc có thể là trung điểm vàng, vừa có tính phi cấu trúc, vừa có các thành phần cấu trúc nhưng ở mức độ hạn chế so với dữ liệu có cấu trúc:
- Nếu tính đến khả năng suy diễn các thành phần cấu trúc từ những văn bản cụ thể, thì trong thực tế hầu như không có văn bản tuyệt đối phi cấu trúc
  - Ví dụ, có thể chia slide này thành 3 mục: Tiêu đề, nội dung, câu hỏi mở rộng
  - Các thành phần cấu trúc có vai trò hữu ích cho những tìm kiếm chính xác, ví dụ kết hợp từ khóa và các thành phần cấu trúc của tài liệu: Tìm các tài liệu trong tiêu đề có C++, và trong mục tác giả có Stroustrup

Tài liệu XML và tìm kiếm XML?

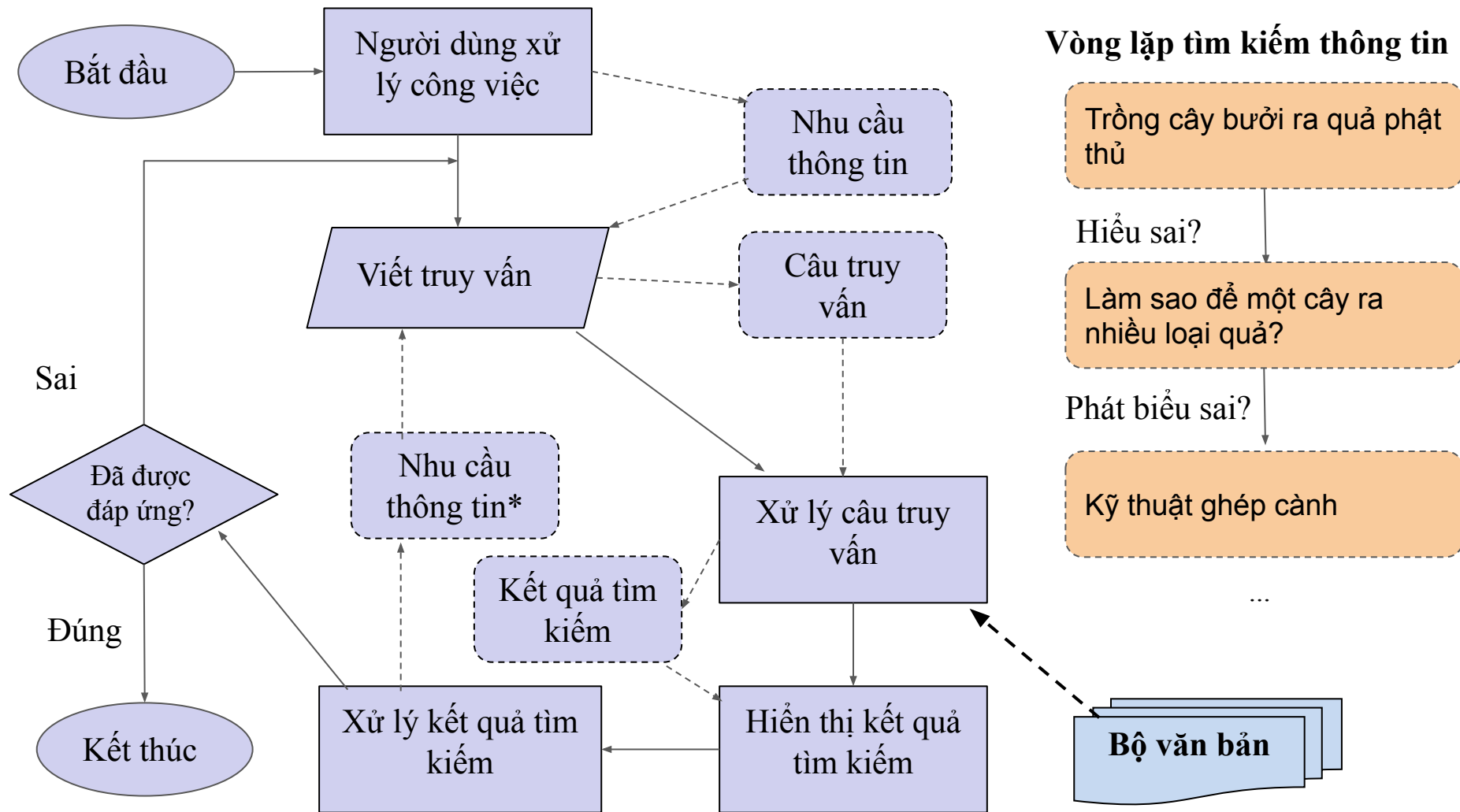
# Giả thuyết cơ bản trong TKTT

Tìm kiếm thông tin cổ điển sử dụng các giả thuyết sau

- **Bộ dữ liệu** là một tập văn bản phi cấu trúc.
  - Ở dạng đơn giản nhất trong máy tính có thể biểu diễn một văn bản phi cấu trúc như một chuỗi ký tự dài.
  - Ở thời điểm này chúng ta thời chưa quan tâm đến các thao tác thay đổi tập văn bản, có thể coi tập văn bản là cố định.
- **Mục đích** của hệ thống TKTT là trích xuất các văn bản có chứa thông tin phù hợp với các nhu cầu thông tin của người dùng nhằm đáp ứng nhu cầu thông tin của người dùng.



# Mô hình tìm kiếm thông tin cổ điển



\*Sau khi nhận kết quả tìm kiếm, người dùng chịu tác động của kết quả tìm kiếm và có thể dẫn đến thay đổi nhu cầu thông tin sau đó thiết lập lại truy vấn.

# Nội dung

1. Vấn đề tìm kiếm thông tin

2. Mô hình Boolean

3. Giải thuật đơn giản đánh chỉ mục trong RAM

4. Xử lý truy vấn Boolean

5. Mô hình Boolean mở rộng và xếp hạng

# Giải pháp Boolean

- Mô hình Boolean có thể được coi là mô hình đơn giản nhất có thể được sử dụng làm cơ sở lý thuyết cho HTTK
- Câu truy vấn trong mô hình Boolean là biểu thức bao gồm các từ và các liên kết lô-gic Boolean
  - Biểu thức truy vấn còn được gọi là biểu thức Boolean trên từ
  - Từ sử dụng để tìm kiếm được gọi là từ khóa
- HTTK trả về tất cả các văn bản thỏa mãn biểu thức Boolean

\*Nhận xét : Gần với hình thức truy xuất dữ liệu

*Câu hỏi: Google được xây dựng dựa trên mô hình Boolean?*

# Ví dụ: Truy vấn Boolean

**Truy vấn:** ((*văn bản* OR *thông tin*) AND *tìm kiếm* AND NOT *lý thuyết*)

**Văn bản:**

1. *Tìm kiếm thông tin*
2. *Lý thuyết thông tin*
3. *Tìm kiếm thông tin hiện đại: lý thuyết và ứng dụng*
4. *Phương pháp nén văn bản*

# Ví dụ 1.1. Truy vấn Boolean (2)

**Truy vấn:** (*văn bản OR thông tin*) AND *tìm kiếm* AND  
NOT *lý thuyết*

**Văn bản:**

1. *Tìm kiếm thông tin*

2. *Lý thuyết thông tin*

3. *Tìm kiếm thông tin hiện đại: lý thuyết và ứng dụng*

4. *Phương pháp nén văn bản*

# Biểu thức truy vấn Boolean

Định nghĩa theo phương pháp quy nạp

|   |                   |  |
|---|-------------------|--|
| 1 | <b>Biểu thức:</b> | $t$ , với $t$ là một từ, là dạng truy vấn đơn giản nhất                  |
|   | <b>Ý nghĩa:</b>   | $R_s = \{d \mid d \in D \text{ và } t \in d\}$                           |
|   | <b>Ví dụ:</b>     | Truy vấn: C++<br>Kết quả: $R_s$ là tập tất cả các văn bản có chứa từ C++ |

|   |                   |  |
|---|-------------------|--|
| 2 | <b>Biểu thức:</b> | NOT $A$ , với $A$ là một biểu thức truy vấn  |
|   | <b>Ý nghĩa:</b>   | $R_s = \overline{R_s(A)}$ , phần bù của tập $R_s(A)$   |
|   | <b>Ví dụ:</b>     | Truy vấn: NOT Java<br>Kết quả: $R_s$ là phần bù của tập văn bản có chứa từ Java, hay nói cách khác là tập hợp các văn bản không chứa từ Java |

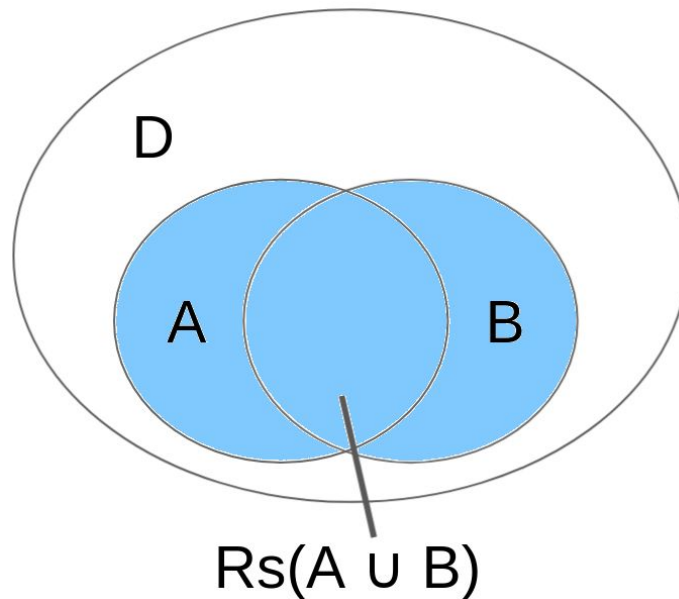
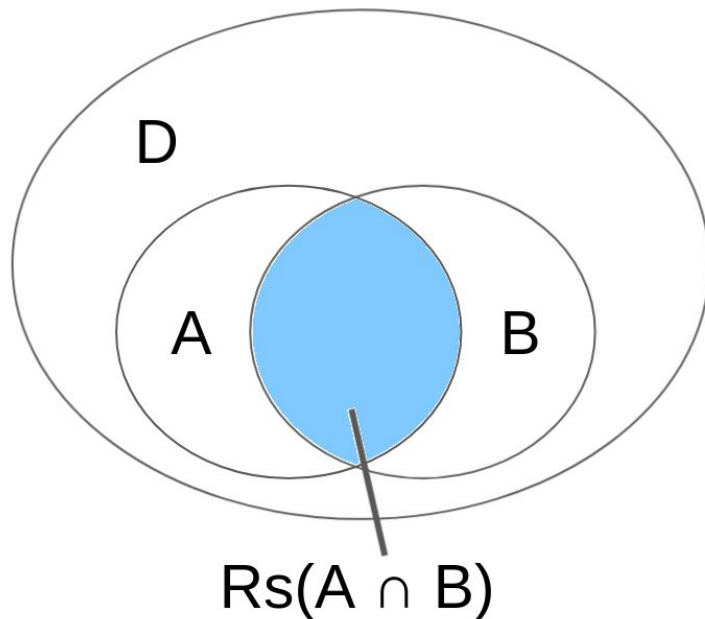
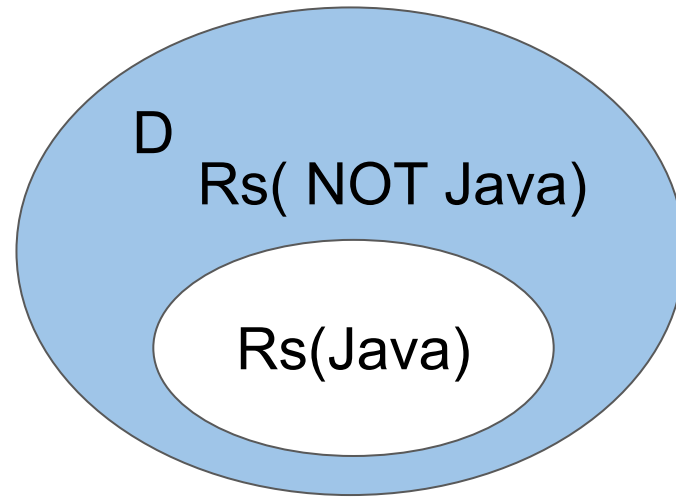
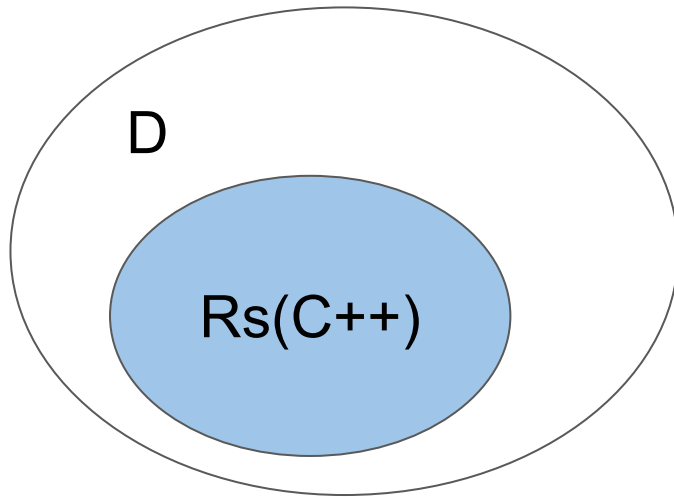
# Biểu thức truy vấn Boolean (2)

Định nghĩa theo phương pháp quy nạp

|   |                   |  |
|---|-------------------|--|
| 3 | <b>Biểu thức:</b> | $A \text{ AND } B$ , với $A, B$ là các biểu thức truy vấn  |
|   | <b>Ý nghĩa:</b>   | $R_S = R_S(A) \cap R_S(B)$   |
|   | <b>Ví dụ:</b>     | Truy vấn: C++ AND Lập trình<br>Kết quả: $R_S$ là giao của tập văn bản có chứa từ C++ và tập văn bản có chứa từ Lập trình |

|   |                   |  |
|---|-------------------|--|
| 4 | <b>Biểu thức:</b> | $A \text{ OR } B$ , với $A, B$ là các biểu thức truy vấn   |
|   | <b>Ý nghĩa:</b>   | $R_S = R_S(A) \cup R_S(B)$   |
|   | <b>Ví dụ:</b>     | Truy vấn: C OR C++<br>Kết quả: $R_S$ là hợp của tập văn bản có chứa từ C và tập văn bản có chứa từ C++ |

# Ví dụ sơ đồ Venn của các tập kết quả truy vấn



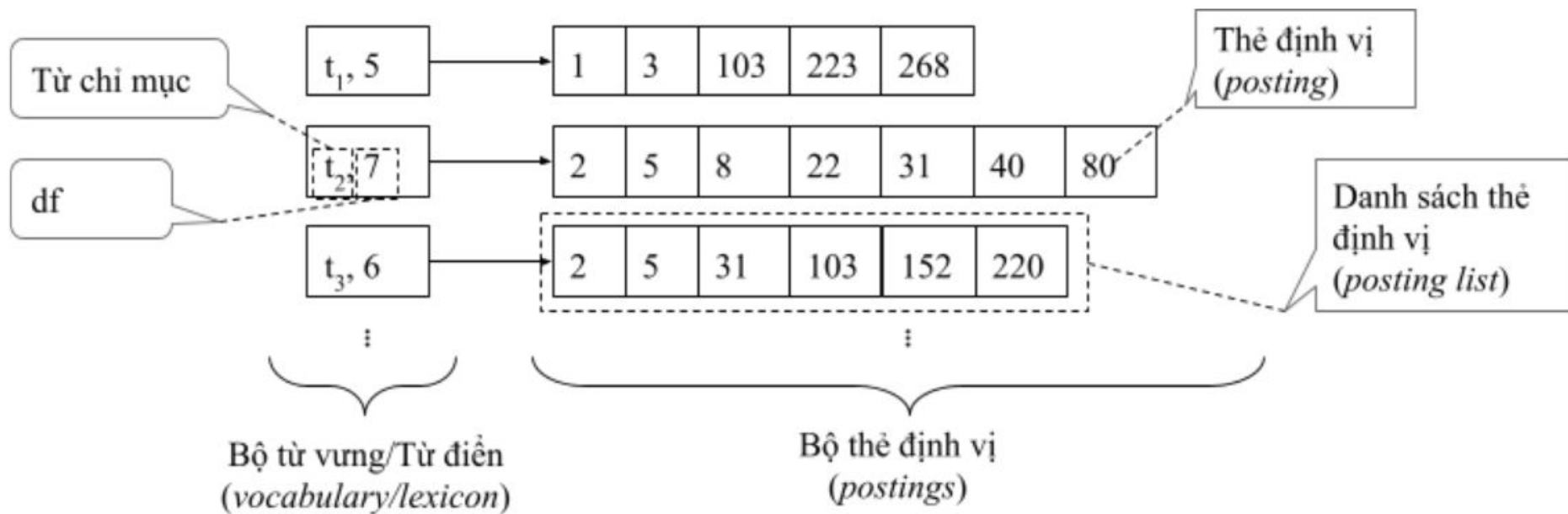


Làm sao để lọc nhanh tập văn bản chứa một từ được cho?

# Chỉ mục ngược

## Chỉ mục ngược (Inverted Index)

là cấu trúc chỉ mục để tra cứu nhanh danh sách các văn bản chứa một từ được cho.



*Ý nghĩa của từ ngược trong chỉ mục ngược?*

# Chỉ mục ngược (2)

Có thể chia chỉ mục ngược thành hai phần có thể lưu trữ tách biệt: Bộ từ vựng và bộ thẻ định vị. Kích thước bộ thẻ định vị có thể lớn hơn nhiều so với bộ từ vựng.

- Bộ từ vựng là một cấu trúc tìm kiếm theo từ:
  - Mỗi mục từ thường là một bộ 3 thành phần: từ (giữ vai trò khóa tìm kiếm), df - số lượng văn bản chứa từ, và con trỏ đến danh sách thẻ định vị tương ứng với mục từ.
- Tất cả danh sách thẻ định vị được gọi gộp là bộ thẻ định vị:
  - Thẻ định vị là một cấu trúc lưu thông tin cho một cặp từ-văn bản (mã văn bản, các vị trí xuất hiện từ, v.v.). Từ định vị mang ý nghĩa xác định vị trí từ xuất hiện trong văn bản;
  - Các thẻ định vị cho một từ được tổ chức thành danh sách có thứ tự để thuận tiện xử lý câu truy vấn.

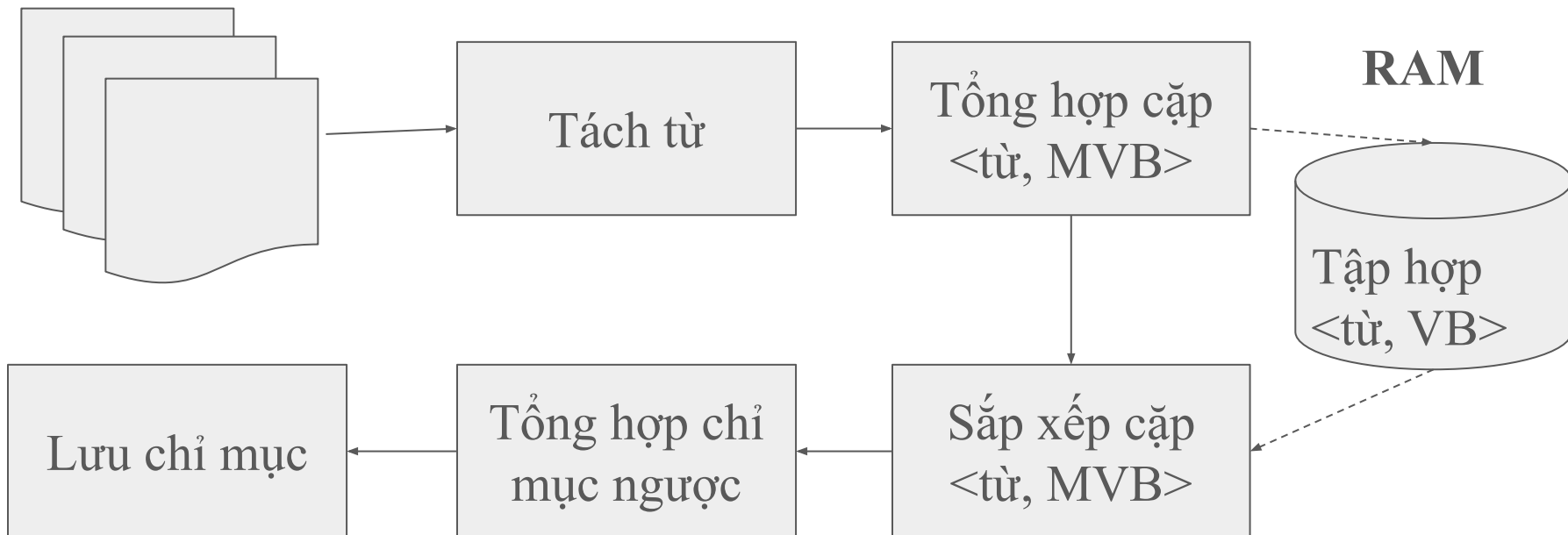
# Nội dung

1. Vấn đề tìm kiếm thông tin
2. Mô hình Boolean
3. Giải thuật đơn giản đánh chỉ mục trong RAM
4. Xử lý truy vấn Boolean
5. Mô hình Boolean mở rộng và xếp hạng

# Giải thuật 1.1. Tạo chỉ mục ngược trong RAM

Chúng ta xét một giải thuật đơn giản để tạo chỉ mục ngược trong RAM

**Yêu cầu:** Bộ nhớ RAM đủ lớn để lưu tất cả dữ liệu phát sinh trong tiến trình xây dựng chỉ mục.

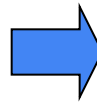


# Ví dụ tạo chỉ mục ngược trong RAM:

## ***B1. Tách từ***

**D1.** DMPLK là tác phẩm văn xuôi đặc sắc và nổi tiếng nhất của Tô Hoài viết về loài vật, dành cho lứa tuổi thiếu nhi

**D2.** Tô Hoài (sinh ngày 27-9-1920) là một nhà văn Việt Nam nổi tiếng. Một số tác phẩm đề tài thiếu nhi của ông được dịch ra ngoại ngữ.



**D1.** DMPKL | là | tác phẩm | văn xuôi | đặc sắc | và | nổi tiếng nhất | của | Tô Hoài | viết về | loài vật | dành cho | lứa tuổi thiếu nhi

**D2.** Tô Hoài | sinh ngày | 27-9-1920 | là một | nhà văn | Việt Nam | nổi tiếng | Một số | tác phẩm | đề tài | thiếu nhi | của ông | được | dịch ra | ngoại ngữ

\*Ghi chú: 1. Giả sử | là ký tự đặc biệt không xuất hiện trong bất kỳ văn bản nào, chúng ta sử dụng | làm dấu hiệu tách từ.

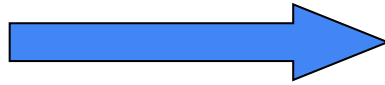
2. DMPLK: Dế mèn phiêu lưu kí

**B2. Tổng hợp cặp  $\langle từ, MVB \rangle$**

**D1.** DMPKL| là | tác phẩm | văn xuôi  
| đặc sắc | và | nổi tiếng nhất | của | Tô  
Hoài | viết về | loài vật | dành cho | lứa  
tuổi thiếu nhi

**D2.** Tô Hoài | sinh ngày | 27-9-1920 |  
là một | nhà văn | Việt Nam | nổi tiếng  
| Một số | tác phẩm | đề tài | thiếu nhi |  
của ông | được | dịch ra | ngoại ngữ

cặp  
<từ, MVB>



| Từ                   | Mã văn bản |
|----------------------|------------|
| <DMPLK,              | 1>         |
| <là,                 |            |
| 1>                   |            |
| <tác phẩm,           | 1>         |
| <văn xuôi,           | 1>         |
| <đặc sắc,            | 1>         |
| <và ,                |            |
| 1>                   |            |
| <nổi tiếng nhất,     | 1>         |
| <của,                | 1>         |
| <Tô Hoài,            | 1>         |
| <viết về,            | 1>         |
| <loài vật,           | 1>         |
| <dành cho,           | 1>         |
| <lứa tuổi thiếu nhi, | 1>         |
| <Tô Hoài,            | 2>         |
| <sinh ngày,          | 2>         |
| <27-9-1920,          | 2>         |
| <là một,             | 2>         |
| <nhà văn,            | 2>         |
| <Việt Nam,           | 2>         |
| <nổi tiếng,          | 2>         |
| <Một số,             | 2>         |
| <tác phẩm,           | 2>         |
| <đề tài,             | 2>         |
| <thiếu nhi,          | 2>         |
| <của ông,            | 2>         |
| <được,               | 2>         |
| <dịch ra,            | 2>         |

## B3. Sắp xếp

| Từ                   | Mã văn bản |
|----------------------|------------|
| <DMPLK,              | 1>         |
| <là,                 | 1>         |
| <tác phẩm,           | 1>         |
| <văn xuôi,           | 1>         |
| <đặc sắc,            | 1>         |
| <và ,                | 1>         |
| <nổi tiếng nhất,     | 1>         |
| <của,                | 1>         |
| <Tô Hoài,            | 1>         |
| <viết về,            | 1>         |
| <loài vật,           | 1>         |
| <dành cho,           | 1>         |
| <lứa tuổi thiếu nhi, | 1>         |
| <Tô Hoài,            | 2>         |
| <sinh ngày,          | 2>         |
| <27-9-1920,          | 2>         |
| <là một,             | 2>         |
| <nhà văn,            | 2>         |
| <Việt Nam,           | 2>         |
| <nổi tiếng,          | 2>         |
| <Một số,             | 2>         |
| <tác phẩm,           | 2>         |
| <đề tài,             | 2>         |
| <thiếu nhi,          | 2>         |
| <của ông,            | 2>         |
| <được,               | 2>         |
| <dịch ra,            | 2>         |
| <ngoại ngữ,          | 2>         |

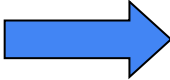
Sắp xếp  
→

| Từ                   | Mã văn bản |
|----------------------|------------|
| <27-9-1920,          | 2>         |
| <DMPLK,              | 1>         |
| <Một số,             | 2>         |
| <Tô Hoài,            | 1>         |
| <Tô Hoài,            | 2>         |
| <Việt Nam,           | 2>         |
| <của,                | 1>         |
| <của ông,            | 2>         |
| <dành cho,           | 1>         |
| <dịch ra,            | 2>         |
| <đặc sắc,            | 1>         |
| <đề tài,             | 2>         |
| <được,               | 2>         |
| <là,                 | 1>         |
| <là một,             | 2>         |
| <loài vật,           | 1>         |
| <lứa tuổi thiếu nhi, | 1>         |
| <ngoại ngữ,          | 2>         |
| <nhà văn,            | 2>         |
| <nổi tiếng,          | 2>         |
| <nổi tiếng nhất,     | 1>         |
| <sinh ngày,          | 2>         |
| <tác phẩm,           | 1>         |
| <tác phẩm,           | 2>         |
| <thiếu nhi,          | 2>         |
| <và ,                | 1>         |
| <văn xuôi,           | 1>         |



| Từ                   | Mã văn bản |
|----------------------|------------|
| <27-9-1920,          | 2>         |
| <DMPLK,              | 1>         |
| <Một số,             | 2>         |
| <Tô Hoài,            | 1>         |
| <Tô Hoài,            | 2>         |
| <Việt Nam,           | 2>         |
| <của,                | 1>         |
| <của ông,            | 2>         |
| <dành cho,           | 1>         |
| <dịch ra,            | 2>         |
| <đặc sắc,            | 1>         |
| <đề tài,             | 2>         |
| <được,               | 2>         |
| <là,                 |            |
| 1>                   |            |
| <là một,             | 2>         |
| <loài vật,           | 1>         |
| <lứa tuổi thiếu nhi, | 1>         |
| <ngoại ngữ,          | 2>         |
| <nhà văn,            | 2>         |
| <nổi tiếng,          | 2>         |
| <nổi tiếng nhất,     | 1>         |
| <sinh ngày,          | 2>         |
| <tác phẩm,           | 1>         |
| <tác phẩm,           | 2>         |
| <thiếu nhi,          | 2>         |
| <và ,                |            |
| 1>                   |            |
| <văn xuôi            | 1>         |

## B4. Tổng hợp<sup>2</sup> chỉ mục ngược


|   |                      |   |      |
|---|----------------------|---|------|
| Tổng hợp<br> | <b>Chỉ mục ngược</b> |   |      |
|   | 27-9-1920, 1         | → | 2    |
|   | ...                  |   |      |
|   | Tô Hoài, 2           | → | 1, 2 |
|   | ...                  |   |      |
|   | tác phẩm, 2          | → | 1, 2 |
|   | ...                  |   |      |
|   | văn xuôi, 1          | → | 1    |
|   | viết về, 1           | → | 1    |

## **B5. Lưu bộ từ vựng và bộ thẻ định vị**

- Bộ từ vựng và bộ thẻ định vị thường được lưu tách biệt
  - Có thể nén chỉ mục ngược để tiết kiệm dung lượng bộ nhớ;
  - Sử dụng các giải thuật khác nhau cho bộ từ vựng và bộ thẻ định vị

Chi tiết các giải thuật nén sẽ được cung cấp sau

# Nội dung

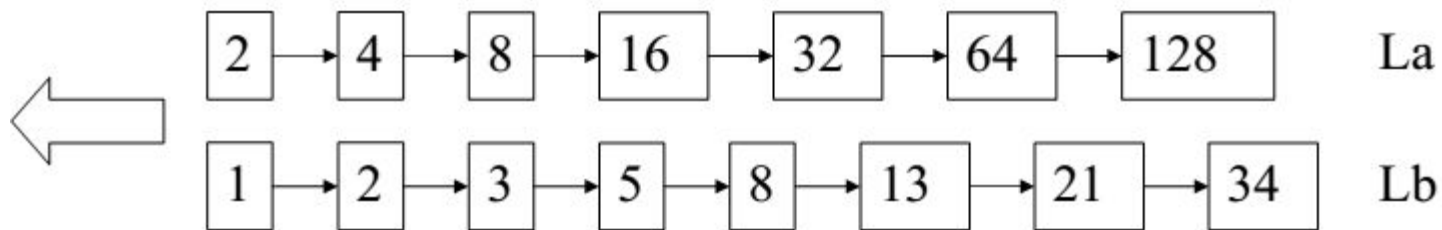
1. Vấn đề tìm kiếm thông tin
  2. Mô hình Boolean
  3. Giải thuật đơn giản đánh chỉ mục trong RAM
  4. Xử lý truy vấn Boolean
  5. Mô hình Boolean mở rộng và xếp hạng
- 

# Truy vấn AND

Các bước thực hiện truy vấn dạng:

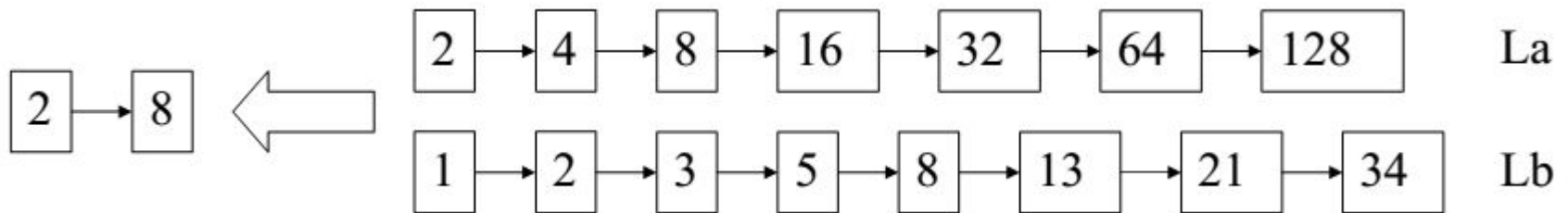
*$a$  AND  $b$ , trong đó  $a, b$  là các từ truy vấn*

1. Tìm  $a$  trong từ điển và lấy danh sách thẻ định vị  $La$
2. Tìm  $b$  trong từ điển và lấy danh sách thẻ định vị  $Lb$
3. Lấy các phần tử chung (giao) của  $La$  và  $Lb$



# Lấy giao của hai danh sách

Thuật toán duyệt đồng thời cả hai danh sách:



Nếu các danh sách được sắp xếp theo mã văn bản, thì số lượng so sánh không vượt quá  $L_a + L_b$ .

## Giải thuật 1.2

```
INTERSECT( $p_1, p_2$ )
1   $answer \leftarrow \langle \rangle$ 
2  while  $p_1 \neq \text{NIL}$  and  $p_2 \neq \text{NIL}$ 
3  do if  $docID(p_1) = docID(p_2)$ 
4      then  $\text{ADD}(answer, docID(p_1))$ 
5           $p_1 \leftarrow next(p_1)$ 
6           $p_2 \leftarrow next(p_2)$ 
7      else if  $docID(p_1) < docID(p_2)$ 
8          then  $p_1 \leftarrow next(p_1)$ 
9          else  $p_2 \leftarrow next(p_2)$ 
10 return  $answer$ 
```

Ví dụ 1.2. Chi tiết các bước thực hiện thuật toán

2, 4, 8, 16, 32, 64, 128    La

1, 2, 3, 5, 8, 13, 21, 34    Lb

*answer* = {2, 8}

| <i>STT</i> | <i>P<sub>a</sub></i> | <i>P<sub>b</sub></i> | <i>answer</i> |
|------------|----------------------|----------------------|---------------|
| 1          | 2                    | 1                    |               |
| 2          | 2                    | 2                    | 2             |
| 3          | 4                    | 3                    |               |
| 4          | 4                    | 5                    |               |
| 5          | 8                    | 5                    |               |
| 6          | 8                    | 8                    | 2, 8          |
| 7          | 16                   | 13                   |               |
| 8          | 16                   | 21                   |               |
| 9          | 32                   | 21                   |               |
| 10         | 32                   | 34                   |               |
| 11         | 64                   | 34                   |               |
| 12         | 64                   | NIL                  |               |

# Xử lý các liên kết Boolean khác

- Liên kết OR có thể được xử lý tương tự như liên kết AND
- Toán tử phủ định (NOT) thường chỉ đi kèm với toán tử AND để giới hạn kích thước tập kết quả, tránh các trường hợp tập kết quả quá lớn.



Trình tự tối ưu thực hiện truy vấn Boolean

# Tối ưu hóa truy vấn toàn AND

**Yêu cầu:** Giải thuật tối ưu hóa phải đủ đơn giản và hiệu quả sao cho tổng thời gian thực hiện tối ưu hóa + xử lý truy vấn theo trình tự tìm được < thời gian xử lý truy vấn theo trình tự thông thường.

**Nhận xét:** Số lượng kết quả của một truy vấn AND không lớn hơn độ dài danh sách thẻ định vị ngắn nhất.

*Một giải thuật tối ưu hóa gần đúng có thể đáp ứng yêu cầu (hữu ích) trong đa số trường hợp:*

1. Tìm các từ truy vấn trong từ điển.
2. Sắp xếp các từ truy vấn theo thứ tự tăng dần độ dài danh sách.
3. Khởi tạo  $R_s$  bằng danh sách đầu tiên (danh sách ngắn nhất)
4. Lần lượt cập nhật  $R_s$  bằng giao của  $R_s$  với các danh sách còn lại theo thứ tự sắp xếp

## Ví dụ 1.3. Tối ưu hóa truy vấn toàn AND:

Cho truy vấn  $a \text{ AND } b \text{ AND } c$  với các danh sách thẻ định vị như trong hình vẽ

Trình tự xử lý thông thường là  $(a \text{ AND } b) \text{ AND } C$

|   |   |   |    |    |    |     |  |    |
|---|---|---|----|----|----|-----|--|----|
| 2 | 4 | 8 | 16 | 32 | 64 | 128 |  | La |
|---|---|---|----|----|----|-----|--|----|

|   |   |   |   |   |    |    |    |    |
|---|---|---|---|---|----|----|----|----|
| 1 | 2 | 3 | 5 | 8 | 16 | 21 | 34 | Lb |
|---|---|---|---|---|----|----|----|----|

|    |    |  |  |  |  |  |  |    |
|----|----|--|--|--|--|--|--|----|
| 13 | 16 |  |  |  |  |  |  | Lc |
|----|----|--|--|--|--|--|--|----|

$df(c) = 2$ ,  $df(a) = 7$ ,  $df(b) = 8$ .

Trình tự thực hiện truy vấn thu được theo giải thuật là:

$(c \text{ AND } a) \text{ AND } b$

*Số bước cho mỗi trình tự thực hiện? Trình tự thu được (sau khi tối ưu hóa) có tối ưu hơn hay không?*

*Trình tự thu được có luôn luôn tối ưu hơn cho mọi truy vấn?*

# AND of OR's

- Ví dụ truy vấn dạng AND of OR's:
  - (văn bản OR dữ liệu OR hình ảnh) AND
  - (nén OR gom nhóm) AND
  - (tìm kiếm OR đánh chỉ mục OR lưu trữ)
- **Nhận xét:** Số lượng kết quả của truy vấn OR không vượt quá tổng số lượng văn bản trong các danh sách thành phần.
- Giải thuật gần đúng tối ưu hóa truy vấn:
  - 1. Tìm các từ truy vấn trong bộ từ vựng.
  - 2. Ước lượng số kết quả cho mỗi truy vấn OR bằng tổng độ dài các danh sách của các từ truy vấn có trong đó.
  - 3. Sắp xếp các truy vấn OR theo thứ tự tăng dần số lượng kết quả
  - 4. Thực hiện các truy vấn theo trình tự sắp xếp:
    - Khởi tạo  $R_s$  là tập kết quả của truy vấn thành phần toàn OR đầu tiên
    - Lần lượt cập nhật  $R_s$  bằng giao của  $R_s$  với kết quả của truy vấn thành phần (toàn OR) tiếp theo theo thứ tự sắp xếp.

*Truy vấn Boolean trong trường hợp tổng quát?*

# Nội dung

1. Vấn đề tìm kiếm thông tin
2. Mô hình Boolean
3. Giải thuật đơn giản đánh chỉ mục trong RAM
4. Xử lý truy vấn Boolean
5. Mô hình Boolean mở rộng và xếp hạng

# Vấn đề mở rộng mô hình Boolean

Nhược điểm cơ bản của mô hình Boolean: Khó kiểm soát tập kết quả (không có cơ chế để cân bằng tính chính xác và tính đầy đủ) vì vậy kém hiệu quả khi áp dụng cho bộ dữ liệu lớn:

- Sử dụng nhiều liên kết AND có xu hướng trả về tập kết quả nhỏ có tính chính xác cao (nhưng có thể bỏ qua nhiều kết quả phù hợp).
- Sử dụng nhiều liên kết OR có xu hướng trả về nhiều kết quả, tập kết quả có tính đầy đủ cao (tuy nhiên có thể có nhiều kết quả không phù hợp gây nhiễu).

## Các giải pháp:

- Bổ xung thêm các cấu trúc tìm kiếm để mở rộng khả năng diễn đạt nhu cầu thông tin;
- Xếp hạng tập kết quả.

# Các cấu trúc tìm kiếm mở rộng

- Các toán tử khoảng cách:
  - /s - ràng buộc các từ xuất hiện trong cùng câu,
  - /p - các từ xuất hiện trong cùng đoạn văn, và
  - /k - các từ xuất hiện trong giới hạn khoảng cách k từ.
- Dấu nháy kép " để tìm kiếm nguyên đoạn văn.
  - Ví dụ: "Lập trình C++": Tìm văn bản có chứa nguyên đoạn "Lập trình C++"
- Các ký tự đại diện
  - Ví dụ, prog!, dấu ! ở cuối từ cho phép khớp với tất cả các từ có cùng phần bắt đầu: program, programming, programmer, v.v...

*Chi tiết các giải thuật xử lý những truy vấn này được cung cấp sau*

# Tìm kiếm có xếp hạng

- Hình thức tìm kiếm thông tin phổ biến nhất hiện nay
  - Đặc biệt là trong môi trường Web
- Người dùng mô tả nhu cầu thông tin bằng một vài từ trong ngôn ngữ tự nhiên (không yêu cầu các toán tử), trong ngữ cảnh TKTT chúng ta gọi là từ truy vấn.
- Hệ thống trả về các kết quả theo một thứ tự ưu tiên dựa trên khả năng đáp ứng nhu cầu thông tin của người dùng
- Về mặt nguyên lý chúng ta có hai lựa chọn truy vấn:
  - Sử dụng các toán tử truy vấn hoặc không
  - Trong thực tế ứng dụng, hình thức tìm kiếm có xếp hạng thường gắn liền với truy vấn dạng văn bản (không chứa các toán tử).



# Đại lượng xếp hạng và hàm giá trị trạng thái tìm kiếm

- Để xếp hạng các văn bản theo một truy vấn  $q$ , với mỗi văn bản  $d$  chúng ta phải tính một đại lượng thể hiện mức độ tương thích với câu truy vấn
  - Tùy thuộc vào mô hình, có thể là độ tương đồng, khoảng cách, xác suất, v.v..
- Chúng ta gọi đại lượng được sử dụng để xếp hạng văn bản là giá trị trạng thái tìm kiếm
  - Hàm giá trị trạng thái tìm kiếm cho văn bản  $d$  và truy vấn  $q$  trả về một giá trị số, ký hiệu là  $RSV(d, q)$ 
    - $RSV = \text{Retrieval Status Value}$
  - Các văn bản được trả về theo thứ tự giảm dần  $RSV$ .

# Nguyên lý xếp hạng kết quả tìm kiếm

*"Trong tìm kiếm có xếp hạng thứ tự tương đối giữa các kết quả tìm kiếm có ý nghĩa quan trọng, trực tiếp ảnh hưởng đến trải nghiệm người dùng và hiệu quả sử dụng hệ thống tìm kiếm, các giá trị trạng thái tìm kiếm cụ thể của các văn bản không quan trọng."*

- Ví dụ sử dụng  $\log x + \log y$  thay cho  $\log (x * y)$
- Các kết quả phù hợp ở ngay đầu danh sách giúp tiết kiệm thời gian của người dùng
- Hệ thống thường không hiển thị các giá trị cụ thể của RSV cho người dùng

Quan trọng đối với các mô hình tìm kiếm có xếp hạng

# Số lượng kết quả tìm kiếm và xếp hạng

- Có thể xử lý vấn đề số lượng lớn kết quả tìm kiếm với tìm kiếm có xếp hạng:
  - Người dùng vẫn có thể nhanh chóng tìm thấy kết quả phù hợp ở ngay đầu danh sách kết quả nếu giải thuật xếp hạng đủ hiệu quả.
- Trong tìm kiếm có xếp hạng danh sách kết quả thường được chia thành các trang 10 kết quả
  - Trang đầu tiên là top  $k$  ( $\approx 10$ ) kết quả được đánh giá cao nhất
  - Người dùng có thể lựa chọn xem thêm các trang tiếp theo
  - $\Rightarrow$  không cung cấp quá nhiều thông tin cho người dùng (có thể tạo cảm giác choáng ngợp) và rút ngắn thời gian phản hồi.

# Bài tập 1.1

Cho các văn bản sau:

- **Doc1:** [Lập trình Hướng đối tượng với C++]
- **Doc2:** [Xây dựng dịch vụ Web với C++]
- **Doc3:** [Kiến trúc nguyên khối và kiến trúc dịch vụ nhỏ]
- **Doc4:** [Kiến trúc hệ thống thông tin]

a) Vẽ biểu diễn chỉ mục ngược;

b) Các văn bản nào sẽ được trả về cho truy vấn:

- Lập trình AND C++
- Xây dựng AND (Dịch vụ OR Hệ thống)

# Bài tập 1.2

Đối với truy vấn toàn AND, thứ tự tăng dần độ dài danh sách thể định vị có luôn là thứ tự tối ưu hay không? Chứng minh?

## Bài tập 1.3

Hãy viết thuật toán thực hiện các truy vấn dạng  $a \text{ OR } b$  và  $a \text{ AND NOT } b$  với độ phức tạp tuyến tính.

Có thể tham khảo thuật toán xử lý truy vấn AND (Giải thuật 1.2)

## Bài tập 1.4

Trong trường hợp tìm kiếm các văn bản tiếng việt, những phát biểu sau đây đúng hay sai?

- a. Trong mô hình tìm kiếm Boolean, loại bỏ dấu không bao giờ làm giảm tính chính xác.
- b. Trong mô hình tìm kiếm Boolean, loại bỏ dấu không bao giờ làm giảm tính đầy đủ.
- c. Loại bỏ dấu làm tăng kích thước bộ từ vựng.
- d. Nên thực hiện các thao tác chuẩn hóa trong quá trình xây dựng chỉ mục thay vì khi thực hiện truy vấn.

# Bài tập 1.5

Hãy xác định trật tự thực hiện truy vấn theo giải thuật tối ưu hóa đã học dựa trên những dữ liệu sau

*(Hoa Đào OR Hoa Mai) AND  
(Quả Bưởi OR Quả Cam) AND  
(Ống Nhòm OR Kính Thiên Văn)*

| Từ truy vấn    | df     |
|----------------|--------|
| Hoa Đào        | 125000 |
| Hoa Mai        | 115000 |
| Quả Bưởi       | 85000  |
| Quả Cam        | 90000  |
| Ống Nhòm       | 218000 |
| Kính Thiên Văn | 112000 |



# Bài tập 1.6

Cho truy vấn:

(Hoa Đào OR Hoa Mai) AND NOT (Nhựa OR Vải)

a) Hãy sử dụng luật phân tích và viết lại truy vấn đã cho dưới dạng OR of ANDs.

b) Trình tự thực hiện truy vấn thu được ở mục a hiệu quả hơn hay kém trình tự ban đầu? Kết luận này đúng trong trường hợp tổng quát? Hay còn phụ thuộc vào từ khóa và các đại lượng thống kê nào khác?

