Tìm kiếm và trình diễn thông tin

Bài 1. Phương pháp tìm kiếm Boolean

Liên lạc với giảng viên

Email: tuyethai@ptithcm.edu.vn

Tiêu đề (Subject): phải bắt đầu với **[2022-2 CD HTTT 01]**

Email không ghi rõ tiêu đề trên sẽ được bộ lọc tự động xoá

Đề cương môn học

Lý thuyết: 15

Chuyên cần (10%) Điểm danh ngẫu nhiên trong lớp

Giữa kỳ (30%) Thi vấn đáp

Cuối kỳ (60%) Thi trắc nghiệm

Tài liệu tham khảo:

An Introduction to Information Retrieval, Christopher D. Manning Prabhakar Raghavan Hinrich Schütze, Cambridge University Press Cambridge, England

Nội dung chính

1. Khái niệm tìm kiếm thông tin
2. Khái niệm mô hình
3. Mô hình Boolean và chỉ mục ngược

Tìm kiếm thông tin là gì?

Tìm kiếm thông tin là tìm kiếm các tài nguyên thông tin phi cấu trúc (thường là văn bản) từ một nguồn thông tin lớn (thường được lưu trên máy tính), đáp ứng được nhu cầu thông tin.

Thuật ngữ tiếng Anh là Information Retrieval (IR).

TKTT vs. CSDL:
Dữ liệu có cấu trúc vs. phi cấu trúc

Dữ liệu có cấu trúc thường thể hiện được dưới dạng bảng

Employee	Manager	Salary
Smith	Jones	50000
Chang	Smith	60000
Ivy	Smith	50000

Cho phép truy xuất dạng so khớp và giới hạn miền giá trị, vd, Salary < 60000 AND Manager = Smith.

TKTT vs. CSDL:
Dữ liệu có cấu trúc vs phi cấu trúc (2)

- Dữ liệu phi cấu trúc: Điển hình là những văn bản tự do.
- Cho phép:
  - Truy xuất bằng từ khóa
    - có thể kết hợp với ràng buộc logic
  - Sử dụng quan hệ ngữ nghĩa giữa các khái niệm, v.d,
    - tìm tất cả những trang web liên quan tới công nghệ

Dữ liệu bán cấu trúc

- Trong thực tế, hầu như rất hiếm dữ liệu văn bản tuyệt đối phi cấu trúc.
  - Nếu tính đến cả khả năng suy diễn cấu trúc yếu từ dữ liệu phi cấu trúc:
    - vd., có thể chia slide này thành hai phần là tiêu đề và nội dung
- Khái niệm bán cấu trúc nằm giữa khái niệm phi cấu trúc và khái niệm có cấu trúc theo mức độ chặt chẽ,
  - Có thể kết hợp phong cách tìm kiếm trên dữ liệu phi cấu trúc và phong cách tìm kiếm trên dữ liệu có cấu trúc cho dữ liệu bán cấu trúc,
  - vd., Tiêu đề có từ thông tin và Nội dung có từ tìm kiếm
  - Tiêu đề nói về lập trình C++ và Tác giả có tên như là stro*rup

Mô hình tìm kiếm thông tin (1)

“Mô hình tìm kiếm là nền tảng lý thuyết để xây dựng công cụ tìm kiếm.”

Nếu biết mô hình được sử dụng để xây dựng công cụ tìm kiếm thì có thể giải thích và dự đoán được hành vi của hệ thống tìm kiếm, v.d., vì sao văn bản A được trả về trước văn bản B? vì sao văn bản C không được trả về? làm thế nào để chiếm thứ hạng cao trong xếp hạng? V.v.

Mô hình tìm kiếm thông tin (2)

Mô hình tìm kiếm quyết định các yếu tố sau:

D: Cách biểu diễn văn bản;

Q: Cách biểu diễn truy vấn;

F: Nền tảng lý thuyết (toán học) tương thích với D và Q, giữ vai trò cơ sở để thực hiện các suy diễn xếp hạng;

R(d, q): Hàm xếp hạng, là hàm định lượng mức độ phù hợp giữa văn bản và truy vấn.

Mô hình Boolean

Ra đời từ khoảng 3 thập kỷ trước đây và là mô hình được sử dụng rộng rãi nhất trong thời gian đó.

D: Văn bản được biểu diễn dưới dạng tập từ;

Q: Biểu thức Boolean trên từ, ràng buộc sự xuất hiện của từ trong văn bản;

F: Lý thuyết tập hợp, đại số Boolean;

R: Một văn bản phù hợp nếu nó thỏa mãn biểu thức truy vấn. R(d, q) chỉ trả về hai giá trị 0: không phù hợp, 1: phù hợp.

Ví dụ phù hợp Boolean

Truy vấn: ((văn bản ∨ thông tin) ∧ tìm kiếm ∧ ¬lý thuyết)

Văn bản:

1. “Tìm kiếm thông tin”

2. “Lý thuyết thông tin”

3. “Tìm kiếm thông tin hiện đại: lý thuyết và thực hành”

4. “Phương pháp nén văn bản”

Ví dụ phù hợp Boolean

Truy vấn: ((văn bản ∨ thông tin) ∧ tìm kiếm ∧ ¬lý thuyết)

Văn bản:

1. “Tìm kiếm thông tin”
2. “Lý thuyết thông tin”
3. “Tìm kiếm thông tin hiện đại: lý thuyết và thực hành”
4. “Phương pháp nén văn bản”

Thực hiện truy vấn Boolean trên dữ liệu nhỏ

- Kiểm tra tuần tự tất cả văn bản:
  - Đơn giản, nhưng…
  - .. Sẽ rất chậm khi chạy trên bộ dữ liệu lớn

Chỉ mục

Khái niệm:

"Index (Chỉ mục) là cấu trúc dữ liệu chuyên biệt để tối ưu hóa tốc độ thực hiện truy vấn."

Ý tưởng sử dụng chỉ mục

Ma trận đánh dấu

1: từ xuất hiện trong văn bản; 0: từ không xuất hiện.

Brutus and Caesar and not Calpurnia: 110100 AND 110111 AND 101111 = 100100

	Anthony and Cleopatra	Julius Caesar	The Tempest	Hamlet	Othello	Macbeth	...
ANTHONY	1	1	0	0	0	1	
BRUTUS	1	1	0	1	0	0	
CAESAR	1	1	0	1	1	1	
CALPURNIA	0	1	0	0	0	0	
CLEOPATRA	1	0	0	0	0	0	
MERCY	1	0	1	1	1	1	
WORSER	1	0	1	1	1	0	
...							
result:	1	0	0	1	0	0	

Xử lý truy vấn trên ma trận đánh dấu

Xử lý các truy vấn Boolean có thể quy về thực hiện phép toán logic theo bit:

Ví dụ, truy vấn a AND b AND NOT d được thực hiện như sau:

Brutus and Caesar and not Calpurnia

1101001 AND 1001101 AND 1011010 = 1001000

Ưu điểm: Nhanh hơn kiểm tra tuần tự;

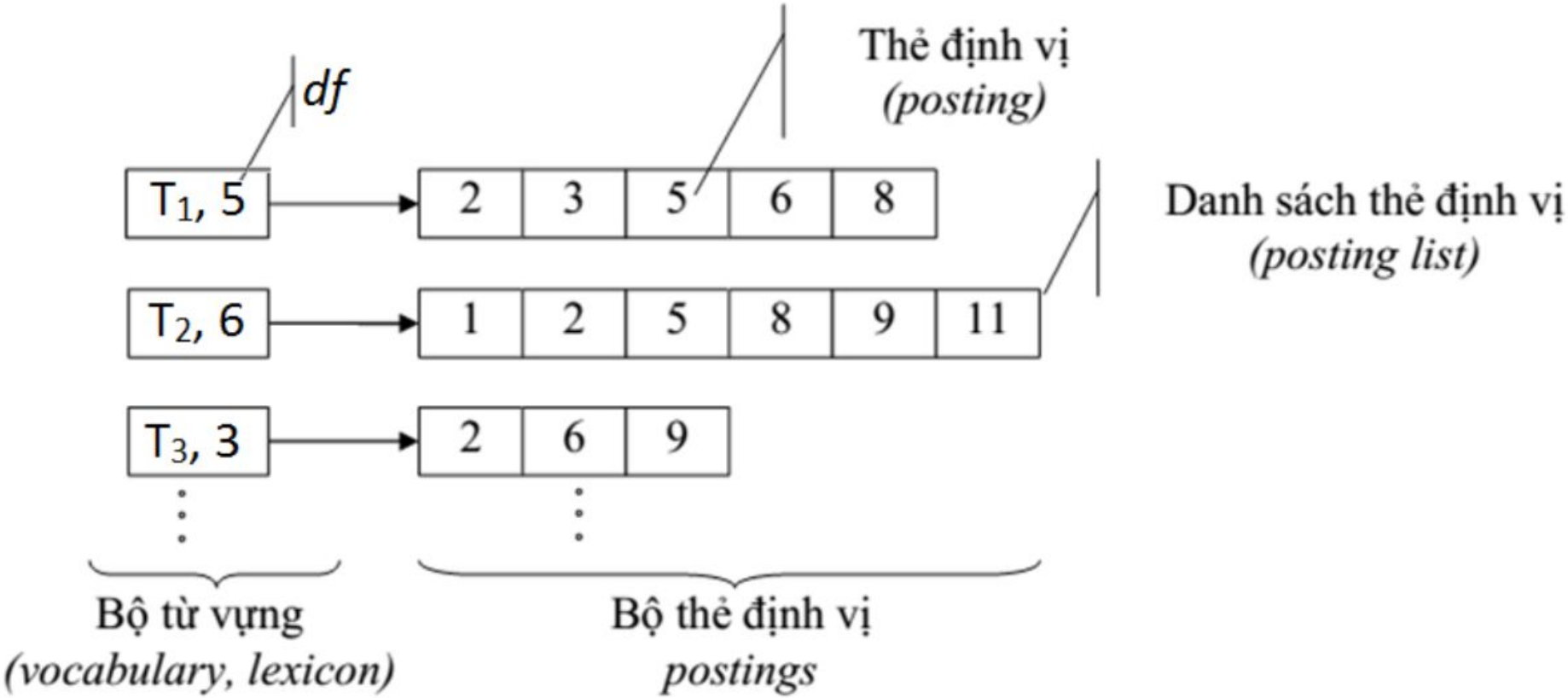
Nhược điểm: nhưng sẽ cần rất nhiều bộ nhớ;

Phương hướng giải quyết nhược điểm: chỉ mục ngược.

Chỉ mục ngược (1)

- Ý tưởng: Gần giống với ma trận đánh dấu, chỉ lưu các giá trị 1.
- Tối ưu hơn ma trận đánh dấu về mặt lưu trữ; Thực hiện truy vấn trên các danh sách:
  - Không thực hiện phép toán logic trên bit như đối với ma trận đánh dấu; Thực hiện các phép toán tập hợp trên danh sách: lấy phần tử chung của hai danh sách ($\cap$), kết hợp hai danh sách ($\cup$);
  - Nếu sắp xếp văn bản theo trật tự tăng dần mã văn bản, thì có thể thực hiện truy vấn với độ phức tạp tuyến tính.

Chỉ mục ngược (2)



Chỉ mục ngược (3)

Các thuật ngữ:

- Mỗi mục từ là một bộ ba gồm một từ duy nhất trong bộ từ vựng, df và con trỏ tới danh sách thẻ định vị của từ đó;
- Thẻ định vị, là một cấu trúc lưu thông tin tương ứng với một văn bản (mã văn bản, các vị trí xuất hiện từ, v.v.). Từ định vị mang ý nghĩa xác định vị trí xuất hiện của từ;
- Tất cả các danh sách thẻ định vị gộp lại được gọi chung là bộ thẻ định vị.

Xây dựng chỉ mục ngược

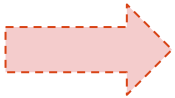
Các bước cơ bản xây dựng chỉ mục ngược trong bộ nhớ:

Tách từ → Sinh thẻ định vị → Sắp xếp thẻ định vị → Tổng hợp danh sách thẻ định vị → Lưu bộ từ vựng và bộ thẻ định vị

Tách từ ….

Doc 1. I did enact Julius Caesar: I was killed in the Capitol; Brutus killed me.

Doc 2. So let it be with Caesar. The noble Brutus hath told you Caesar was ambitious:



Doc 1. i did enact julius caesar i was killed in the capitol brutus killed me

Doc 2. so let it be with caesar the noble brutus hath told you caesar was ambitious

Sinh thẻ định vị

Doc 1. i did enact julius caesar i was killed i' the capitol brutus killed me

Doc 2. so let it be with caesar the noble brutus hath told you caesar was ambitious

term	docID
i	1
did	1
enact	1
julius	1
caesar	1
i	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1

term	docID
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

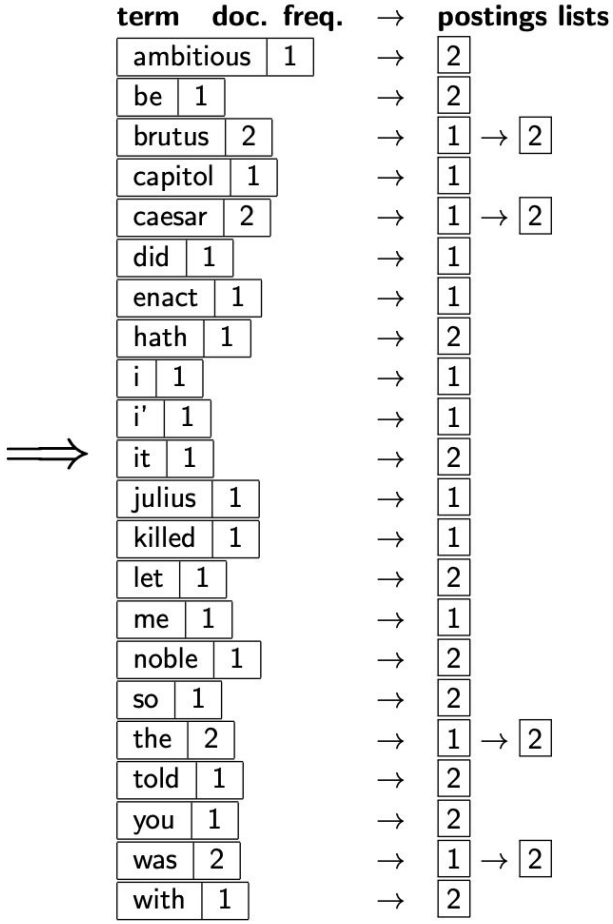
Sắp xếp thẻ định vị

term	docID
i	1
did	1
enact	1
julius	1
caesar	1
i	1
was	1
killed	1
i'	1
the	1
capitol	1
brutus	1
killed	1
me	1
so	2
let	2
it	2
be	2
with	2
caesar	2
the	2
noble	2
brutus	2
hath	2
told	2
you	2
caesar	2
was	2
ambitious	2

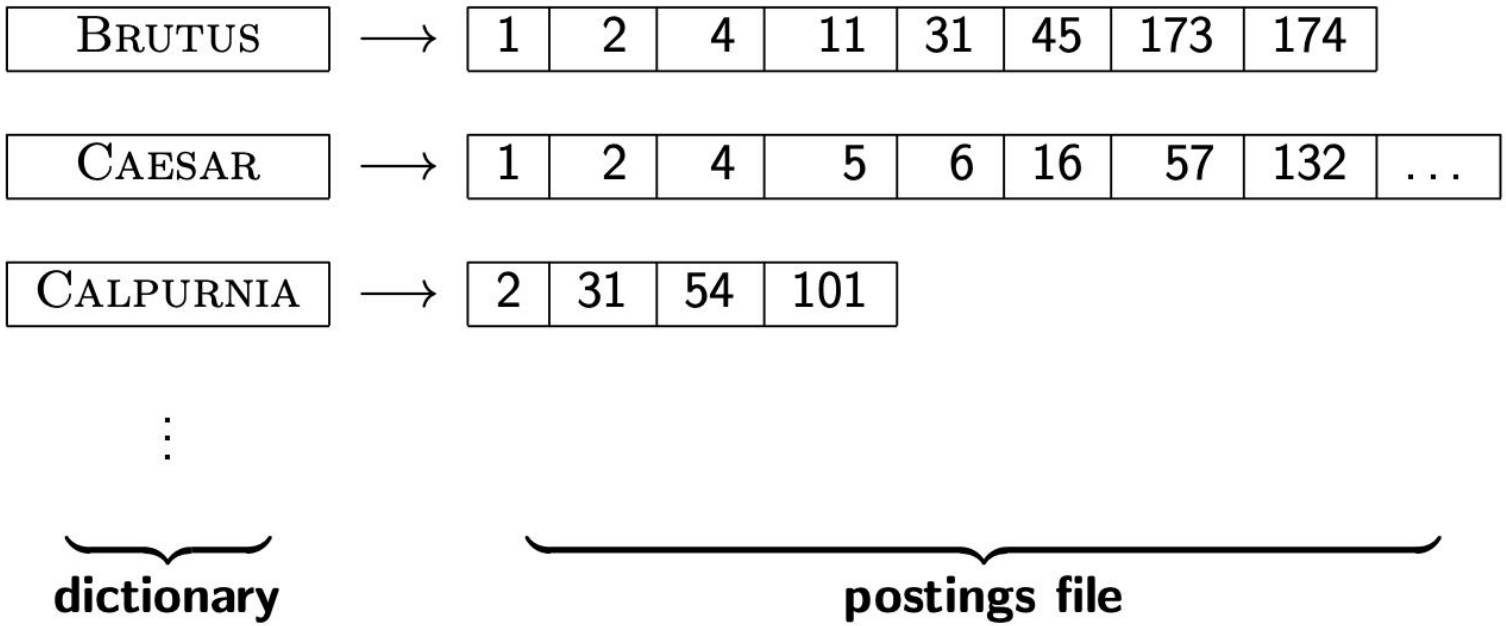
term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
i	1
i	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2

Tổng hợp danh sách thẻ định vị

term	docID
ambitious	2
be	2
brutus	1
brutus	2
capitol	1
caesar	1
caesar	2
caesar	2
did	1
enact	1
hath	1
i	1
i	1
i'	1
it	2
julius	1
killed	1
killed	1
let	2
me	1
noble	2
so	2
the	1
the	2
told	2
you	2
was	1
was	2
with	2



Lưu bộ từ vựng và bộ thẻ định vị



Bài tập

Cho các văn bản sau:

Doc1: [breakthrough drug for schizophrenia]

Doc2: [new schizophrenia drug]

Doc3: [new approach for treatment of schizophrenia]

Doc4: [new hopes for schizophrenia patients]

a) Vẽ biểu diễn chỉ mục ngược;

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

schizophrenia AND drug

for AND NOT(drug OR approach)

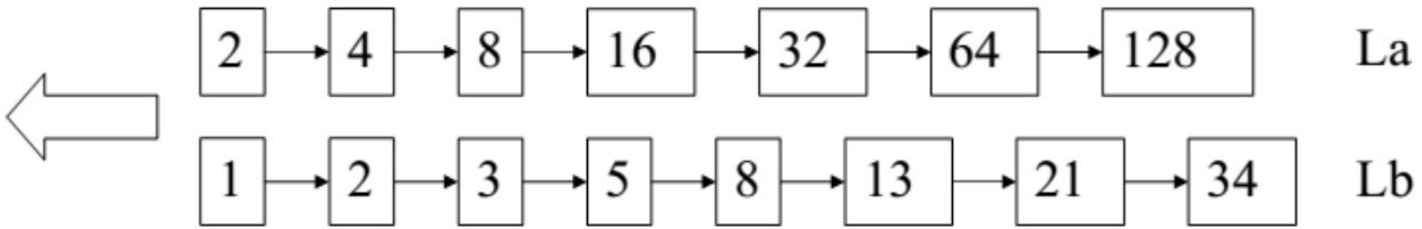
Thực hiện truy vấn trên bộ chỉ mục ngược

Các bước thực hiện truy vấn kiểu: a b -> a and b

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



Thực hiện truy vấn trên bộ chỉ mục ngược

Lấy giao của hai danh sách

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

Thực hiện truy vấn trên bộ chỉ mục ngược

2, 4, 8, 16, 32, 64, 128 La

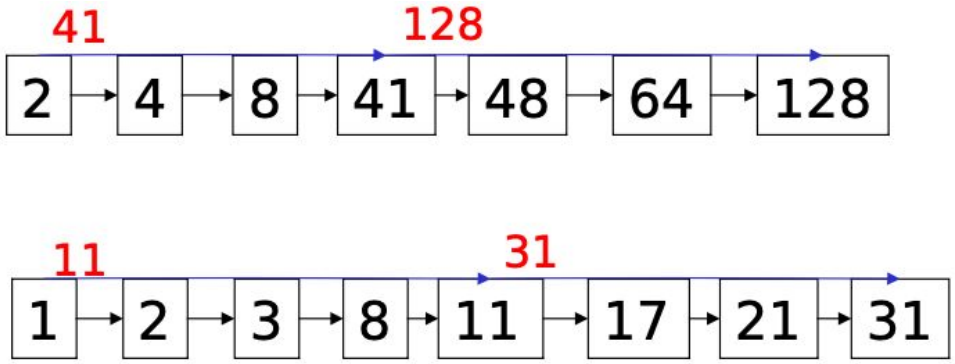
1, 2, 3, 5, 8, 13, 21, 34 Lb

```
INTERSECT(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then p1 ← next(p1)
 9                else p2 ← next(p2)
10  return answer
```

La	Lb	answer
2	1	
2	2	2
4	3	
4	5	
8	5	
8	8	2, 8
16	13	
16	21	
32	21	
32	34	
64	34	
64	NIL	

Cải tiến giải thuật lấy giao hai danh sách

- Bổ sung bước nhảy vào danh sách thẻ định vị;
- Sử dụng bước nhảy để bỏ qua những thẻ định vị không thỏa mãn điều kiện.

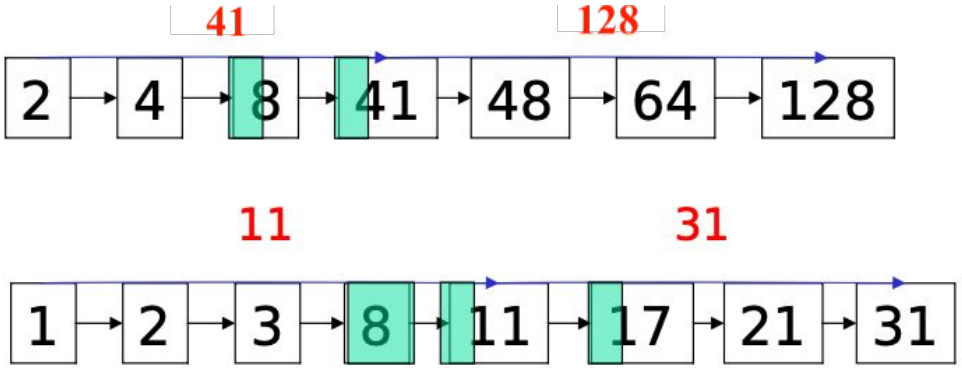


Cải tiến giải thuật lấy giao hai danh sách

```
INTERSECTWITHSKIPS(p1, p2)
 1  answer ← ⟨ ⟩
 2  while p1 ≠ NIL and p2 ≠ NIL
 3  do if docID(p1) = docID(p2)
 4        then ADD(answer, docID(p1))
 5             p1 ← next(p1)
 6             p2 ← next(p2)
 7        else if docID(p1) < docID(p2)
 8                then if hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
 9                        then while hasSkip(p1) and (docID(skip(p1)) ≤ docID(p2))
10                                do p1 ← skip(p1)
11                        else p1 ← next(p1)
12                else if hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
13                        then while hasSkip(p2) and (docID(skip(p2)) ≤ docID(p1))
14                                do p2 ← skip(p2)
15                        else p2 ← next(p2)
16  return answer
```

Lấy giao hai danh sách có bước nhảy

- Giả sử trong quá trình duyệt danh sách, các con trỏ đang ở vị trí số 8 ở cả hai danh sách, các thao tác là:
  - Lưu giá trị 8 và,
  - Dịch chuyển con trỏ sang phải ở cả hai danh sách, vị trí mới là (41, 11), Thực hiện bước nhảy (vì 31 < 41), và kết thúc giải thuật.
  - Trong trường hợp này chúng ta đã bỏ qua một phần danh sách



Độ dài của bước nhảy

- Nếu nhiều bước nhảy khoảng cách nhỏ xác suất di chuyển theo bước nhảy cao. Nhưng phải so sánh bước nhảy nhiều lần.
- Ít bước nhảy ít so sánh hơn, nhưng khoảng cách lớn hơn xác suất di chuyển theo bước nhảy thấp hơn.

Tối ưu hóa truy vấn AND

- **Số kết quả không lớn hơn độ dài danh sách thẻ định vị ngắn nhất**

  Query: t1, t2, t3, …

  1. Với mỗi thuật ngữ truy vấn t

     Tìm t trong bộ từ vựng

  2. Sắp xếp thuật ngữ tăng dần theo df(t): **df(t1)** > df(t2) > df(t3)

  3. Khởi tạo tập kết quả answer là danh sách ngắn nhất

  4. Tiếp tục thực hiện truy vấn theo thứ tự đã sắp xếp

  posting_list (t3) and posting_list (t2) and posting_list (t1)

Tối ưu hóa truy vấn AND

Ví dụ: Cho truy vấn a AND b AND c với các danh sách thẻ định vị như trong hình vẽ

2	4	8	16	32	64	128		La
1	2	3	5	8	16	21	34	Lb
13	16							Lc

Thứ tự tối ưu với truy vấn a AND b AND c là (c AND a) AND b

Tối ưu hóa truy vấn AND của OR

- 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)
- Tối ưu hóa truy vấn
  - Lấy độ dài danh sách thẻ vị trí cho mỗi từ
  - Ước lượng số kết quả cho mỗi truy vấn OR
  - Sắp xếp các truy vấn OR theo thứ tự tăng dần số lượng kết quả