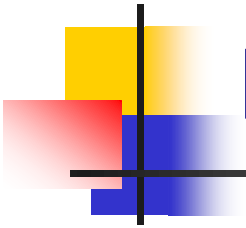
IT4853
Tìm kiếm và trình diễn thông tin

Bài 12. Các phương pháp xây dựng chỉ mục ngược
IIR.C4. Index Construction

TS. Nguyễn Bá Ngọc, *Bộ môn Hệ thống thông tin, Viện CNTT & TT*
ngocnb@soict.hust.edu.vn

Hà Nội, 2016



Nội dung chính

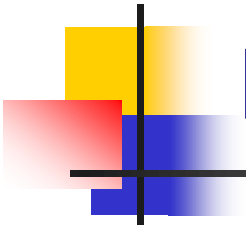
- Phần cứng căn bản
- Các giải thuật xây dựng chỉ mục ngược:
  - BSBI
  - SPIMI
  - MapReduce
- Quản lý bộ dữ liệu động

Phần cứng căn bản

- Tốc độ trao đổi dữ liệu (đọc/ghi) trong bộ nhớ RAM nhanh hơn nhiều so với trên ổ đĩa;
- Không thể trao đổi dữ liệu với ổ đĩa khi đang định vị đầu đọc;
- Thời gian đọc/ghi nguyên một khối dữ liệu (block) hoặc một lượng nhỏ hơn là như nhau;
  - Kích thước khối được xác định trong quá trình định dạng ổ đĩa, phổ biến là 8, 16, 32, 64 Kb.
- BUS hệ thống điều khiển trao đổi dữ liệu giữa ổ đĩa và bộ nhớ RAM. Không sử dụng CPU.

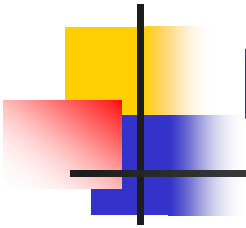
Các đặc trưng phần cứng cơ bản

Ký hiệu	Đặc trưng	Giá trị
s	Thời gian định vị đầu đọc	5 ms = 5 x 10^{-3} s
b	Thời gian trung bình đọc/ghi 1 byte	0.02 µs = 2 x 10^{-8} s
	Chu kỳ đồng hồ bộ vi xử lý	10^{-9} s
p	Thời gian thực hiện một lệnh cơ bản	0.01 µs = 10^{-8} s



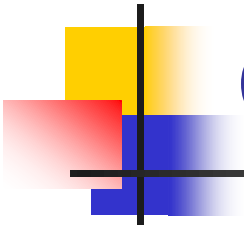
Kích thước chỉ mục

- Giải thuật xây dựng chỉ mục trong bộ nhớ chính chỉ phù hợp với những bộ dữ liệu nhỏ.
- Đối với những bộ dữ liệu lớn, kích thước chỉ mục có thể vượt quá dung lượng của bộ nhớ chính:
  - Cần sử dụng ổ đĩa cứng, hoặc hơn nữa là phân tán chỉ mục trên nhiều máy.



Nội dung chính

- Phần cứng căn bản
- Các giải thuật xây dựng chỉ mục ngược
  - BSBI
  - SPIMI
  - MapReduce
- Quản lý bộ dữ liệu động



Giải thuật BSBI

- Các thao tác cơ bản:
  - Đọc dữ liệu, tách từ và sinh thẻ định vị;
  - Tích lũy thẻ định vị thành khối kích thước xác định;
  - Sinh chỉ mục cho khối và lưu tạm thời trên ổ đĩa;
  - Hợp nhất các chỉ mục khối thành chỉ mục bộ dữ liệu.

Xây dựng chỉ mục dựa trên sắp xếp theo khối: BSBI: Blocked Sort-Based Indexing

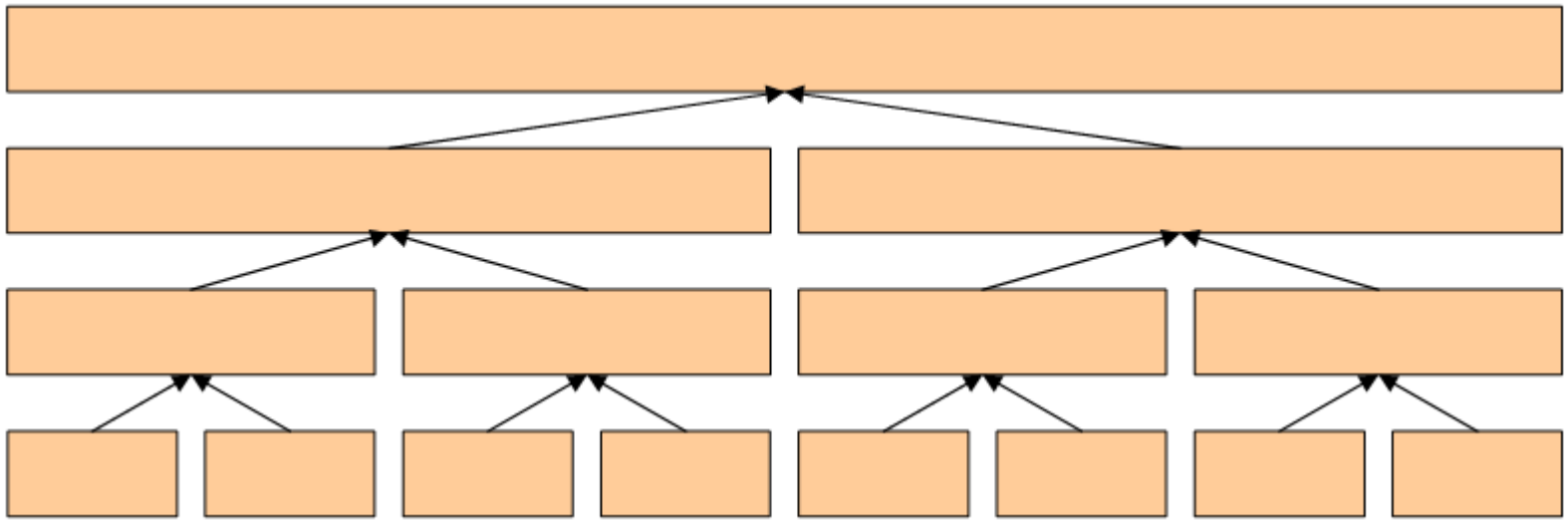
Giải thuật BSBI (2)

BSBIndexConstruction()

```
1  n ← 0
2  while (all documents have not been processed)
3  do n ← n + 1
4       block ← ParseNextBlock()
5       BSBI-Invert(block)
6       WriteBlockToDisk(block, f_n)
7  MergeBlocks(f_1, ..., f_n; f_merged)
```

Hợp nhất chỉ mục

- Sơ đồ hợp nhất theo cặp:

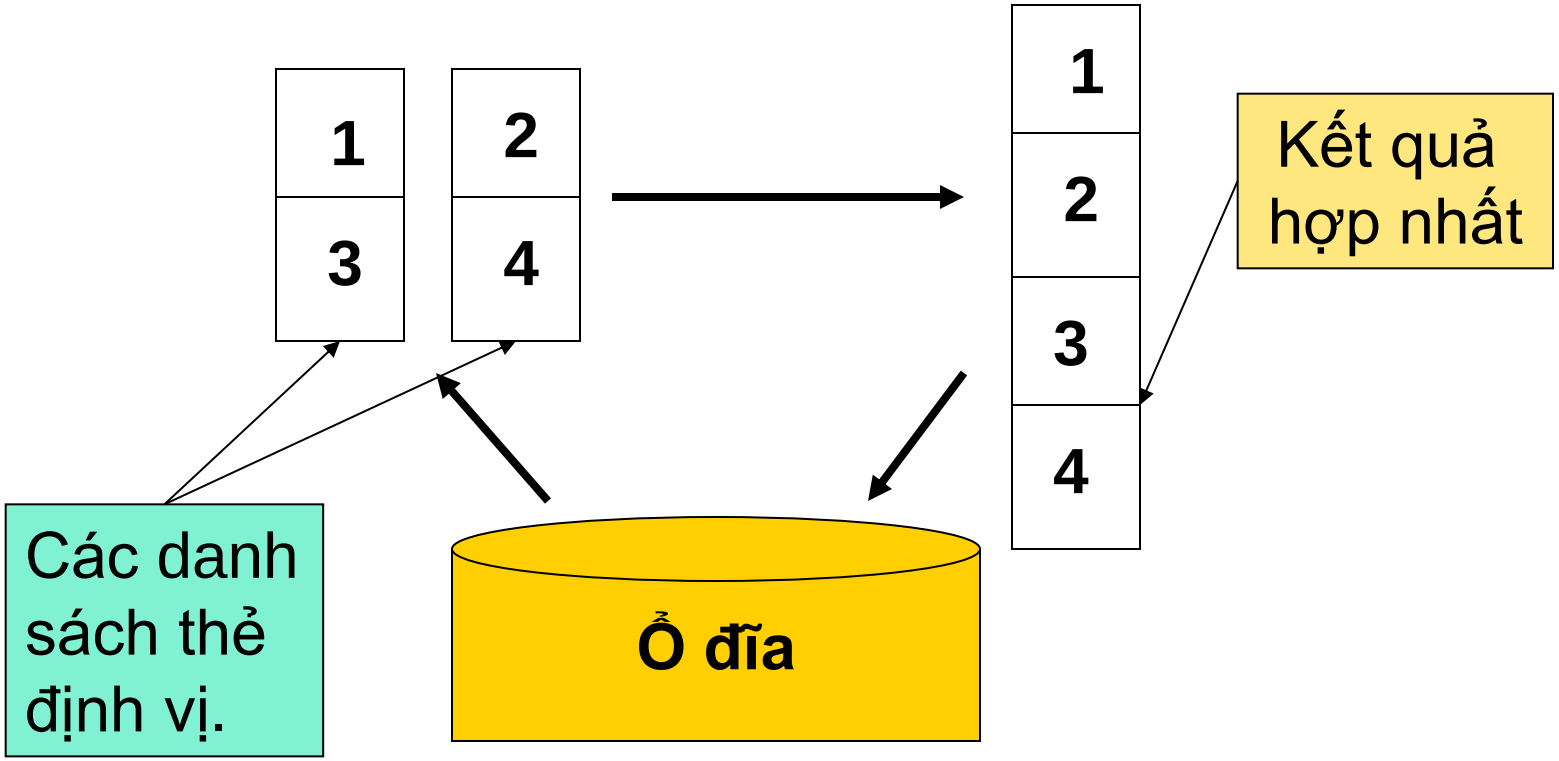


Hợp nhất chỉ mục (2)

Phương pháp hợp nhất đồng thời:

1. Sử dụng bộ nhớ đệm cho từng khối;
2. Đọc dữ liệu vào bộ nhớ đệm từ các tệp tạm thời;
3. Tại mỗi bước lựa chọn từ nhỏ nhất và hợp nhất tất cả các danh sách thẻ định vị, nạp thêm dữ liệu vào bộ nhớ đệm nếu cần;
4. Lưu kết quả hợp nhất lên đĩa.

Hợp nhất danh sách thẻ định vị



Ví dụ hợp nhất chỉ mục

- Khối 1

  t1: 1, 3, 5

  t2: 2, 6, 8

- Khối 2

  t2: 3, 5

  t3: 2, 3, 5

- Kết quả hợp nhất

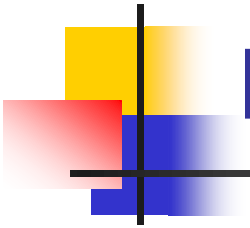
  t1: 1, 3, 5

  t2: 2, 3, 5, 6, 8

  t3: 2, 3, 5

Các hạn chế của BSBI

- Nếu sử dụng <termId,docId>:
  - Cần lưu toàn bộ từ điển trong bộ nhớ chính;
    - Để chuyển đổi từ thành mã từ.
- Nếu sử dụng <term, docId>:
  - Không cần lưu toàn bộ từ điển, nhưng...
  - ...Dữ liệu tạm thời sẽ lớn, làm giảm tốc độ xây dựng chỉ mục;



Nội dung chính

- Phần cứng căn bản
- Các giải thuật xây dựng chỉ mục ngược:
  - BSBI
  - SPIMI
  - MapReduce
- Quản lý bộ dữ liệu động

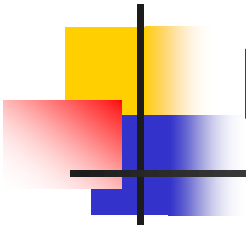
Giải thuật SPIMI

- Sử dụng từ điển riêng biệt cho từng khối;
  - Không cần cung cấp mã từ duy nhất trên toàn bộ dữ liệu;
  - Không cần lưu từ điển đầy đủ cho bộ dữ liệu trong bộ nhớ.
- Thêm trực tiếp thẻ định vị vào danh sách
  - Không cần thực hiện sắp xếp danh sách thẻ định vị.

Xây dựng chỉ mục một lượt trong bộ nhớ chính: SPIMI: Single-pass in-memory indexing;
Chúng ta có thể xây dựng chỉ mục ngược đầy đủ cho mỗi khối;
Sau đó có thể hợp nhất các chỉ mục con lại thành một chỉ mục.

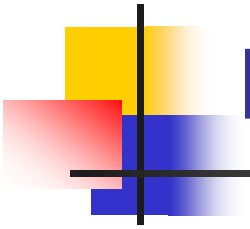
Giải thuật SPIMI (2)

```
SPIMI-INVERT(token_stream)
 1  output_file = NEWFILE()
 2  dictionary = NEWHASH()
 3  while (free memory available)
 4  do token ← next(token_stream)
 5     if term(token) ∉ dictionary
 6        then postings_list = ADDTODICTIONARY(dictionary, term(token))
 7        else postings_list = GETPOSTINGSLIST(dictionary, term(token))
 8     if full(postings_list)
 9        then postings_list = DOUBLEPOSTINGSLIST(dictionary, term(token))
10     ADDTOPOSTINGSLIST(postings_list, docID(token))
11  sorted_terms ← SORTTERMS(dictionary)
12  WRITEBLOCKTODISK(sorted_terms, dictionary, output_file)
13  return output_file
```



Nội dung chính

- Phần cứng căn bản
- Các giải thuật xây dựng chỉ mục ngược:
  - BSBI
  - SPIMI
  - MapReduce
- Quản lý bộ dữ liệu động

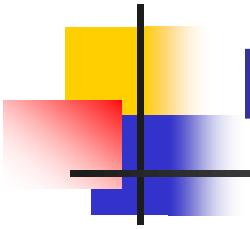


MapReduce

- MapReduce (Dean and Ghemawat 2004) là một kiến trúc tính toán phân tán:
  - Đơn giản: Không cần lập trình tương tác giữa các nút để phân chia công việc, trao đổi dữ liệu, v.v.
  - Độ tin cậy cao: Đảm bảo tính kết thúc trên hệ thống máy tính sử dụng phần cứng phổ thông.

Pha Map: Đọc dữ liệu

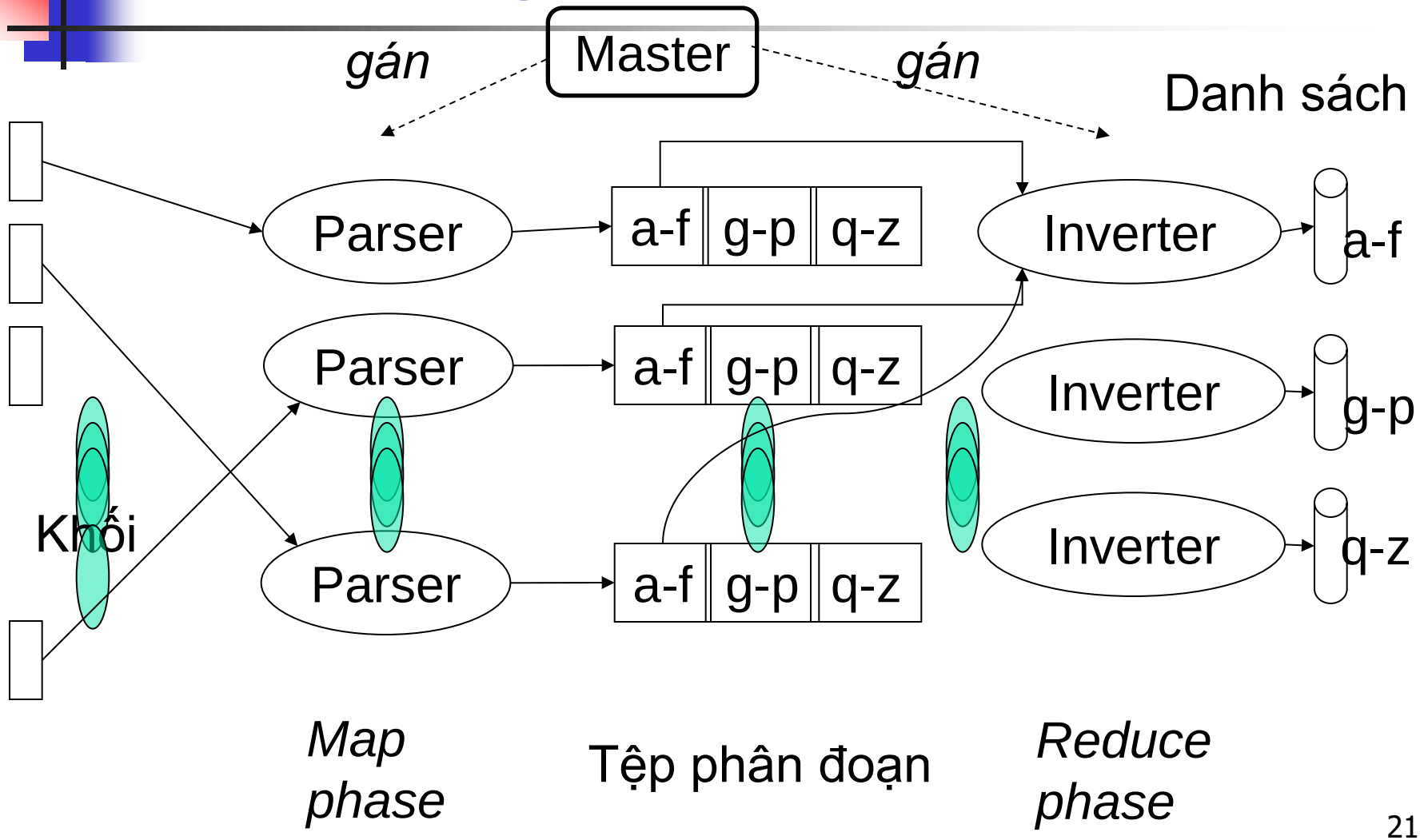
- Nút điều khiển thực hiện phân chia công việc đọc dữ liệu:
  - Chia bộ dữ liệu thành nhiêu khối và phân chia cho các nút đọc dữ liệu;
  - Nút đọc xử lý tuần tự từng văn bản và sinh thẻ định vị, vd, theo dạng cặp <từ, mã văn bản>
  - Sau đó phân chia thẻ định vị vào j phân đoạn: Mỗi phân đoạn ứng với một khoảng từ (ví dụ, j = 3, ứng với ba khoảng: ***a-f, g-p, q-z***).



Pha Reduce: Nghịch đảo

- Số lượng nút nghịch đảo bằng số lượng phân đoạn j;
- Nhiệm vụ của nút nghịch đảo:
  - Tiếp nhận tất cả các phân đoạn tương ứng thu được sau khi đọc dữ liệu;
  - Sắp xếp và thiết lập danh sách thẻ định vị.

Sơ đồ luồng dữ liệu



Các phương pháp phân tán chỉ mục

- *Phân tán theo từ:* mỗi máy xử lý một khoảng từ
- *Phân tán theo văn bản:* mỗi máy xử lý một tập con của bộ văn bản văn bản
- Hầu hết công cụ tìm kiếm sử dụng chỉ mục phân đoạn theo văn bản.
  - Có thể chuyển đổi chỉ mục phân tán theo từ thành chỉ mục phân tán theo văn bản.

Ví dụ xây dựng chỉ mục theo MapReduce

- Map:

  d1 : C ca, C ce.

  d2 : C d.

  →

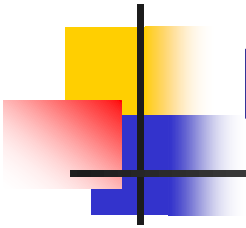
  <C,d1>, <ca,d1>, <C,d1>, <ce, d1>, <C, d2>, <d,d2>

- Reduce:

  (<C,(d1,d2,d1)>, <d,(d2)>, <ca,(d1)>, <ce,(d1)>)

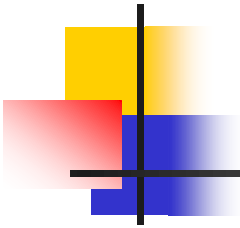
  →

  (<C,(d1:2,d2:1)>, <d,(d2:1)>, <ca,(d1:1)>, <ce,(d1:1)>)



Nội dung chính

- Phần cứng căn bản
- Các giải thuật xây dựng chỉ mục ngược:
  - BSBI
  - SPIMI
  - MapReduce
- Quản lý bộ dữ liệu động



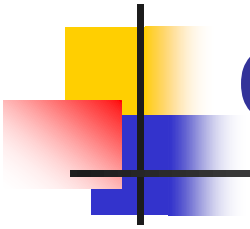
Bộ dữ liệu tĩnh

- Bộ dữ liệu tĩnh: Là bộ dữ liệu không thay đổi hoặc rất ít khi thay đổi.
- Đối với bộ dữ liệu tĩnh khi có sự thay đổi chỉ cần thực hiện xây dựng lại chỉ mục:
  - Cập nhật lại bộ từ vựng;
  - Cập nhật lại danh sách thẻ định vị.

Đối với những bộ dữ liệu thay đổi thường xuyên cần phải có giải pháp quản lý hiệu quả hơn.

Các thao tác quản lý chỉ mục động

- Xóa (delete). Thường chỉ thực hiện xóa ảo vì xóa thực đòi hỏi xây dựng lại chỉ mục.
  - Đánh dấu văn bản muốn xóa;
  - Lọc những văn bản đã đánh dấu khỏi danh sách kết quả.
- Cập nhật (update). Thường được thực hiện thông qua hai thao tác: xóa và thêm mới.
- Thêm mới (insert). Có nhiều phương pháp với độ phức tạp khác nhau để cho phép thêm mới văn bản vào chỉ mục.



Chỉ mục chính phụ

- Sử dụng chỉ mục chính và chỉ mục phụ:
  - Thêm văn bản mới vào chỉ mục phụ;
- Thực hiện truy vấn trên cả hai chỉ mục và tổng hợp kết quả.
- Định kỳ xây dựng lại toàn bộ chỉ mục.

Nhược điểm của giải pháp chỉ mục chính phụ

- Nếu bộ dữ liệu thay đổi rất thường xuyên, thì kích thước chỉ mục phụ có thể tăng nhanh.
  - Cần nhiều thời gian để hợp nhất chỉ mục chính và chỉ mục phụ;
    - **Giải pháp:** Sử dụng nhiều chỉ mục có thể giảm thời gian hợp nhất chỉ mục (**tuy nhiên thực hiện truy vấn sẽ phức tạp hơn*).

```
LMERGEADDTOKEN(indexes, Z_0, token)
 1  Z_0 ← MERGE(Z_0, {token})
 2  if |Z_0| = n
 3     then for i ← 0 to ∞
 4          do if I_i ∈ indexes
 5               then Z_{i+1} ← MERGE(I_i, Z_i)
 6                    (Z_{i+1} is a temporary index on disk.)
 7                    indexes ← indexes − {I_i}
 8               else I_i ← Z_i     (Z_i becomes the permanent index I_i.)
 9                    indexes ← indexes ∪ {I_i}
10                    BREAK
11          Z_0 ← ∅
```

```
LOGARITHMICMERGE()
1  Z_0 ← ∅     (Z_0 is the in-memory index.)
2  indexes ← ∅
3  while true
4  do LMERGEADDTOKEN(indexes, Z_0, GETNEXTTOKEN())
```

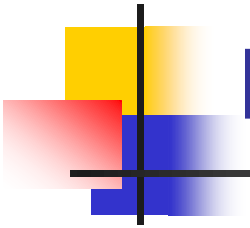
Hợp nhất chỉ mục với độ phức tạp Logarithm

- Sử dụng nhiều cấp chỉ mục:
  - Lưu chỉ mục nhỏ nhất (Z_0) trong bộ nhớ;
  - Khi Z_0 trở nên quá lớn, sẽ ghi Z_0 lên đĩa và thực hiện hợp nhất với những chỉ mục đã tồn tại.
    - Các chỉ mục trên đĩa ký hiệu là I_0, I_1, ..., I_n
    - Các chỉ mục tạm trên đĩa ký hiệu là Z_1, Z_2, ..., Z_n

Bài tập 12.1

Cho n = 2, và 1 <= T <= 30, hãy thực hiện giải thuật LogarithMerge và vẽ bảng thể hiện ở mỗi thời điểm khi T = 2 * k từ đã được xử lý (1 <= k <= 15), các chỉ mục nào trong số bốn chỉ mục I_0...I_4 được sử dụng. Phần đầu của bảng như sau:

	I_3	I_2	I_1	I_0
2	0	0	0	0
4	0	0	0	1
6	0	0	1	0



Bài tập 12.2

Chỉ mục phụ có thể ảnh hưởng đáng kể đến chất lượng thống kê trên bộ dữ liệu. Ví dụ điển hình là idf, được định nghĩa như sau $\log(N/df_i)$ trong đó N là số lượng văn bản và df_i là số văn bản chứa từ thứ i. Hãy chứng minh rằng một chỉ mục phụ dù nhỏ cũng có thể gây sai lệch lớn đến idf nếu chỉ tính trên chỉ mục chính.

Gợi ý: xét một từ hiếm nhưng đột ngột xuất hiện thường xuyên.