TECHNISCHE UNIVERSITÄT DRESDEN

Dresden Database Systems Group

Exercise 1

Information Retrieval

2. Boolean Retrieval, Phrase Queries and Positional Indexes

Term-Document Incidence Matrix

Exercise 2.1

- Consider these documents

  - Doc 1: `the new home sales top the forecasts`
  - Doc 2: `home sales rise in july`
  - Doc 3: `increase in home sales in july`
  - Doc 4: `july new home sales rise`

- Draw the term-document incidence matrix for this document collection ok

- Why can this data structure not be used in practice, i.e., for large document collections? next

in ppt p10

Inverted Index

Exercise 2.2

ok

- Draw the inverted index corresponding to the term-document incidence matrix from exercise 2.1

- Provide the results of the following queries on that inverted index: ok

`forecasts`	`forecasts AND new`
`sales AND NOT home`	`(increase OR rise) AND july`

- Why should postings lists be sorted?

?- in order to intersection

OR-Queries

Exercise 2.3

A postings **intersection** algorithm for queries of the form `x AND y` was presented in the lecture

Write a postings **merge** algorithm for queries of the form `x OR y` in the same style

Intersection of two postings lists

```
INTERSECT($p_1, p_2$)
answer ← ⟨ ⟩
while $p_1 \neq$ NIL and $p_2 \neq$ NIL
do if $docID(p_1) = docID(p_2)$
     then ADD($answer, docID(p_1)$)
          $p_1 \leftarrow next(p_1)$
          $p_2 \leftarrow next(p_2)$
     else if $docID(p_1) < docID(p_2)$
            then $p_1 \leftarrow next(p_1)$
            else $p_2 \leftarrow next(p_2)$
return $answer$
```

Union of two postings lists

ok

NOT-Queries

Exercise 2.4

would be to calculate (NOT
y) first as a new postings list, which takes O(N

- How should the Boolean query `x AND NOT y` be handled?
- Why is the naive evaluation of this query normally very expensive?
- Write out a postings merge algorithm that evaluates this query efficiently.

ok
照片中那个
书后答案有，其实和merge很像

- For the queries below, can we still run through the intersection in time $O(|x| + |y|)$, where $|x|$ and $|y|$ are the lengths of the postings lists for `x` and `y`? If not, what can we achieve?

`x AND NOT y`

O(x+y)

`x OR NOT y`

O(N)

++(Brutus OR Caesar) AND NOT (Anthony OR Cleopatra) 书后习题

Query Processing Order

Dresden Database
Systems Group

Exercise 2.5

- Recommend a query processing order for the query ok

```
(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
```

- Given the following postings list sizes:

Term	Postings size
eyes	213 312
kaleidoscope	87 009
marmalade	107 913

Term	Postings size
skies	271 658
tangerine	46 653
trees	316 812

Query Processing Order

Exercise 2.5

- Recommend a query processing order for the query

```
(tangerine OR trees) AND (marmalade OR skies) AND (kaleidoscope OR eyes)
```

- Given the following postings list sizes:

Term	Postings size
eyes	213 312
kaleidoscope	87 009
marmalade	107 913

Term	Postings size
skies	271 658
tangerine	46 653
trees	316 812

- For a conjunctive query, is processing postings lists in order of size guaranteed to be optimal?
- Explain why it is, or give an example where it is not.

Exercise 0.9-- NOT!!!

Skip Pointers

Exercise 2.6

Exercise 0.19

- We have a two-word AND-query with the following corresponding postings lists:
  - [4, 6, 10, 12, 14, 16, 18, 20, 22, 32, 47, 81, 120, 122, 157, 180] (P=16 entries)
  - [47]
- How many comparisons would be done to intersect the two postings lists with the following two strategies?
  - Using standard postings lists ok
  - Using postings lists stored with skip pointers, with a skip length of $\sqrt{P}$ as suggested in the lecture
- Can skip pointers be used for OR-queries? If so, how?

it is essential to visit every docID in the posting lists of either terms, thus killing the need for skip pointers

Bi-Word Index

Exercise 2.7

- How many vocabulary terms does the bi-word index corresponding to the inverted index from exercise 2.2 have?

10 ?

- Consider the phrase query “`dresden's finest restaurant`”
- Give an example of a (short) document which is a false positive when this query is run over a bi-word index, i.e., as “`dresden's finest`” AND “`finest restaurant`”

Document=”Some alumni had arrived from New York. University faculty
said that Stanford is the best place to study....”.

Positional Inverted Index (1)

Exercise 2.8

Exercise 0.23

- Enrich the inverted index from exercise 2.2 with position information
- Provide the results of the following queries, whereby `a /n b` means `b` at most `n` tokens after `a`

`home /1 sales`

`sales /2 july`

Positional Inverted Index (2)

Exercise 0.22

Exercise 2.9

ok

- Shown below is a portion of a positional index in the format:
  term : doc1: ‹pos1, pos2, …› ; doc2: ‹pos1, pos2, …› ; …

angels :	2: ‹36, 174, 252, 651› ;	4: ‹12, 22, 102, 432› ;	7: ‹17›
fools :	1: ‹1, 17, 74, 222› ;	4: ‹8, 78, 108, 458› ;	7: ‹3, 13, 23, 193›
fear :	2: ‹87, 704, 722, 901› ;	4: ‹13, 43, 113, 433› ;	7: ‹18, 328, 528›
in :	2: ‹3, 37, 76, 444, 851› ;	4: ‹10, 20, 110, 470, 500› ;	7: ‹5, 15, 25, 195›
rush :	3: ‹2, 66, 194, 321, 702› ;	4: ‹9, 69, 149, 429, 569› ;	7: ‹4, 14, 404›
to :	2: ‹47, 86, 234, 999› ;	4: ‹14, 24, 774, 944› ;	7: ‹199, 319, 599, 709›
tread :	2: ‹57, 94, 333› ;	4: ‹15, 35, 155› ;	8: ‹20, 320›
where :	2: ‹67, 124, 393, 1001› ;	4: ‹11, 41, 101, 421, 431› ;	9: ‹16, 36, 736›

- Which document(s) if any meet each of the following queries, where each expression within quotes is a phrase query?

“fools rush in”

“fools rush in” AND “angels fear to tread”