Information Organization
Exercises Part 1 (Information Retrieval)

I-1

Suppose that there is an inverted index holding terms and their document frequencies shown below:

Kitakyushu: 20052 documents, **Wakamatsu**: 3152 documents, **Orio**: 12052 documents

1. Consider query (a) **Kitakyushu** AND **Wakamatsu** AND **Orio**. Show an efficient way of processing query (a).
2. Suppose that term **Kokura** and term **Hakata** have document frequencies x and y, respectively. Shown an algorithm to answer query (b) **Kokura** AND NOT **Hakata**, that runs in $O(x + y)$ time. Also, briefly justify the time complexity.

I-2

1. Explain why block-based merge is more efficient than quick sort for disk-based index construction.
2. Explain n-way merge algorithm ($n > 2$) using illustrations.

I-3

Suppose that the following document IDs are in a postings list.

10, 14, 29, 134, 149, 153

Show a compact representation of the above postings list, using bits as few as possible. How many bits are necessary?

I-4

Consider building a dictionary of the following seven terms:

append appendix appetizer applause apple application applied

1. Show a dictionary where the dictionary is stored as a long string of characters.
2. Assume that four bytes are used for a pointer to a term. How many bytes are used for the dictionary of (1)?
3. Assume that each term in the dictionary is queried equally likely. Then what is the average number of terms compared?

4. Now show a dictionary where blocking with four terms is used. How many bytes are used for the 4-term blocked dictionary? Here assume that one byte is used to store a term length. Also, what is the average number of terms compared for this dictionary?

5. Show a dictionary where front coding is used. Now how many bytes are necessary?

I-5

1. Suppose that we count the frequency of each term within a document collection, and rank the terms by their frequencies in a decreasing order. What kind of relationship holds between the ranks and term frequencies? What is the name of the empirical law that explains this relationship?

2. Now we construct posting lists from a document collection that satisfies the above law. Then, how the law affects the length of each posting list? How you estimate the total size of all the posting lists?

I-6

Suppose that there is a collection of $N = 1,000,000$ documents, with document frequencies (df) of terms

tomato, cheese, pizza mushroom, fish

are 10000, 10000, 50000, 5000, 1000, respectively.

Now compute the tf-idf score between the query **fish cheese pizza** and the document **tomato cheese pizza cheese pizza**, by filling the table below. For tf-idf score, use ltn.lnc, namely logarithm term frequency $(1 + \log(tf_{t,d}))$, idf $(\log N/df_t)$ document frequency, no normalization for the query, and logarithm term frequency, no document frequency, and cosine normalization for the document. Use $\log 2 = 0.30$ for approximation.

term	query					document			product
	tf	wf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
tomato									
cheese									
pizza									
mushroom									
fish									

I-7

Suppose that there is a collection of $N = 1,000,000$ documents, with document frequencies (df) of terms

red, yellow, apple, orange

are 50000, 10000, 20000, 5000, respectively.

Now compute the tf-idf score between the query **red apple** and the document **red apple orange**, by filling the table below. For tf-idf score, use ltn.lnc, namely logarithmic term frequency $(1+\log(tf_{t,d}))$, idf $(\log N/df_t)$ document frequency, no normalization for the query, and logarithmic term frequency, no document frequency, and cosine normalization for the document. You can use $\log 2 = 0.30$ for approximation.

term	query					document			product
	tf	wf	df	idf	$w_{t,q}$	tf	wf	$w_{t,d}$	
red									
yellow									
apple									
orange									

I-8

Consider the following documents d_1, d_2 and d_3:

d_1: apple tomato apple apple orange
d_2: orange potato potato tomato apple
d_3: apple peach lemon potato

1. We want to rank the terms in the documents by importance, using tf-idf score. Calculate the tf-idf score of each term, using ltn.lnc, namely logarithmic term frequency $(1+\log_{10}(tf_{t,d}))$, idf $(\log_{10} N/df_t)$ document frequency. Here, $N = 3$, and use $\log_{10} 1.5 = 0.18, \log_{10} 2 = 0.30, \log_{10} 3 = 0.48$.

2. We want to measure similarity among d_1, d_2 and d_3, by cosine similarity. (1) Show the term frequency vectors of d_1, d_2 and d_3, where the term frequency is logarithmic $(1+\log(tf_{t,d}))$, and length-normalized. (2) Compute the cosine values for each pair of the three vectors, and determine which document pair is most similar.

I-9

The table below shows the results of human judges A and B marking either relevant (1) or non-relevant (0) to documents with doc ID from 1 to 10, for a query Q. Also, an IR system W retrieved documents for the same query Q (marked 1 for retrieved, 0 for not retrieved).

doc ID	1	2	3	4	5	6	7	8	9	10
human judge A	1	0	0	1	1	0	1	0	0	1
human judge B	1	1	0	1	0	0	1	0	1	1
IR system W	1	0	0	0	0	1	1	0	1	1

1. Calculate precision, recall, and F_1 score for IR system W if a document is considered relevant only when the two judges agree.

2. Calculate precision, recall, and F_1 score for IR system W if a document is considered relevant if either judge thinks it is relevant.

3. Estimate how much degree the two judges agree on query Q.

I-10

The table below shows whether documents $d_1, \ldots, d_{10}$ are either relevant ('R') or non-relevant ('N') to a certain query Q.

document	d_1	d_2	d_3	d_4	d_5	d_6	d_7	d_8	d_9	d_{10}
result	R	N	N	R	N	R	N	R	R	N

Now consider two search systems SysA and SysB. These systems returned the following ranked results for the same query Q:

rank	1	2	3	4	5	6	7	8	9	10
SysA	d_4	d_5	d_6	d_7	d_8	d_1				
SysB	d_6	d_4	d_8	d_5						

The above results mean the followings: SysA returned six documents as relevant documents, and among them SysA ranked d_4 as No. 1. The documents missing from SysA's results were judged by SysA as non-relevant. The same applies to SysB.

1. Calculate precision, recall, and F_1 measure of SysA.

2. Calculate precision, recall, and F_1 measure of SysB.

3. Calculate precision at rank 4 for both SysA and SysB.

4. Discuss what the results of above (1)-(3) imply.

I-11

Dicuss what query expansion techniques are used in web search engines.

I-12

1. What is a snippet of a query result? Describe several examples of snippets of recent search engines you know.

2. Why snippets are useful?

3. What kind of techniques are necessary for realizing snippets?

I-13

1. Two IR systems IR-A and IR-B are evaluated on a query over 100 documents, and the following results are obtained.

Result of IR-A

	relevant	nonrelevant
retrieved	21	7
not retrieved	52	20

Result of IR-B

	relevant	nonrelevant
retrieved	40	27
not retrieved	12	21

   Compare characteristics of IR-A and IR-B by computing precision, recall and F_1 measure.

2. Argue what would be a good way of evaluating search engines which return a large number of documents, but users are interested only in highly-ranked documents.

I-14

1. Suppose that System A and System B for information retrieval were evaluated on the same document collection, over query to find "ramen recipe". The results were as follows:

```
System A:  Retrieved  80 documents, in which  20 documents were about
ramen recipe.  All the documents about ramen recipe in the collection
were retrieved.

System B:  Retrieved 20 documents, in which 15 documents were about
ramen recipe.
```

   Now compare qualities of System A and System B using appropriate measures.

2. In web search, potentially all the web pages in the Internet can be the document collection. Why recall is difficult to measure in this situation? What is a good measure to evaluate quality of retrieved results in web search?