

Seminar 1

Exercise 1

Recommend a query processing strategy for (*tangerine* OR *trees*) AND (*marmalade* OR *skies*) AND (*kaleidoscope* OR *eyes*) with respect to the following postings list sizes:

eyes 213312

kaleidoscope 87009

marmalade 107913

skies 271658

tangerine 46653

trees 316812

We use a database trick where we filter out the results with the clause of the shortest intermediate result first. Operations OR is understood as addition and AND as multiplication. Compose the equations:

$$\textit{tangerine} \text{ OR } \textit{trees} = 46653 + 316812 = 363465$$

$$\textit{marmalade} \text{ OR } \textit{skies} = 107913 + 271658 = 379571$$

$$\textit{kaleidoscope} \text{ OR } \textit{eyes} = 87009 + 213312 = 300321$$

After sorting these with respect to sizes and we get the ordering

$$\textit{kaleidoscope} \text{ OR } \textit{eyes} < \textit{tangerine} \text{ OR } \textit{trees} < \textit{marmalade} \text{ OR } \textit{skies}$$

we see that the query is best processed in the following sequence:

1. $a = \textit{kaleidoscope} \text{ OR } \textit{eyes}$
2. $b = \textit{tangerine} \text{ OR } \textit{trees}$
3. $c = \textit{marmalade} \text{ OR } \textit{skies}$
4. $d = a \text{ AND } b$
5. $e = d \text{ AND } c$

Exercise 2

Create an inverted index composed of the following collection of documents:

Doc 1: new home sales top forecasts

Doc 2: home sales rise in July

Doc 3: increase in home sales in July

Doc 4: July new home sales rise

Very easy procedure. Start with an empty table. If the term already appears in the table as a key, add the document ID only. Otherwise, take each term of a document and add it as a key to the table with the ID of the document. This way we get the inverted index represented in the following table.

new	1	4		
home	1	2	3	4
sales	1	2	3	4
top	1			
forecasts	1			
rise	2	4		
in	2	3		
July	2	3	4	
increase	3			

Table 1: Inverted index