Exercise 1.2 (IIR book)

- Consider these documents
  - Doc1: breakthrough drug for schizophrenia
  - Doc2: new schizophrenia drug
  - Doc3: new approach for treatment of schizophrenia
  - Doc4: new hopes for schizophrenia patients
- A: Draw the term-document incidence matrix for this document collection
- B: Draw the inverted index representation for this collections, as in slide 13 (part 2)
Exercise 1.3

- For the document collection shown in Exercise 1.2, what are the returned results for these queries
  - A: schizophrenia AND drug
  - B: for AND NOT (drug OR approach)
  - C: for AND (NOT drug) AND (NOT approach)
Exercise 1.4

- Adapt the merge for the queries:
  - *Brutus AND NOT Caesar*
  - *Brutus OR NOT Caesar*
- Can we still run through the merge in time $O(x + y)$?
- If not, what can we achieve?
Exercise 1.5

- What about an arbitrary Boolean formula? 
  \((\text{Brutus OR Caesar}) \ AND \ NOT\ (\text{Antony OR Cleopatra})\)

- Can we always merge in “linear” time?
  - Linear in what?

- Can we do better?
Exercise

- Compute the complexity of the general merge/intersect algorithm

\textsc{Intersect}(\langle t_1, \ldots, t_n \rangle)
1. \( \text{terms} \leftarrow \text{SortByIncreasingFrequency}(\langle t_1, \ldots, t_n \rangle) \)
2. \( \text{result} \leftarrow \text{postings}(\text{first(terms)}) \)
3. \( \text{terms} \leftarrow \text{rest(terms)} \)
4. \textbf{while} \text{terms} \neq \text{NIL} \text{ and } \text{result} \neq \text{NIL}
5. \textbf{do} \text{result} \leftarrow \text{Intersect}(\text{result}, \text{postings}(\text{first(terms)}))
6. \text{terms} \leftarrow \text{rest(terms)}
7. \textbf{return} \text{result}
Exercise 1.7

- Recommend a query processing order for

\[(\text{tangerine OR trees}) \ AND \ (\text{marmalade OR skies}) \ AND \ (\text{kaleidoscope OR eyes})\]

<table>
<thead>
<tr>
<th>Term</th>
<th>Freq</th>
</tr>
</thead>
<tbody>
<tr>
<td>eyes</td>
<td>213312</td>
</tr>
<tr>
<td>kaleidoscope</td>
<td>87009</td>
</tr>
<tr>
<td>marmalade</td>
<td>107913</td>
</tr>
<tr>
<td>skies</td>
<td>271658</td>
</tr>
<tr>
<td>tangerine</td>
<td>46653</td>
</tr>
<tr>
<td>trees</td>
<td>316812</td>
</tr>
</tbody>
</table>
Exercise 1.8

- If the query is:
  - **friends** AND **romans** AND (NOT **countrymen**)  
- how could we use the freq of countrymen in evaluating the best query evaluation order?
- Propose a way of handling negation in the INTERSECT algorithm
- Then propose a way for determining the order of query processing.
Exercise 1.9

For a conjunctive query, is processing postings list in order of size guaranteed to be optimal? Explain why it is, or give an example where it isn't.
Exercise

- Try the search feature at http://www.rhymezone.com/shakespeare/
- Write down five search features you think would also be useful
Exercise 2.5, 2.6

2.5: Why are skip pointers not useful for queries of the form $x \text{ OR } y$?

2.6: We have a two-word query. For one term the postings list consists of the following 16 entries:
- \[4,6,10,12,14,16,18,20,22,32,47,81,120,122,157,180\]

and for the other it is the one entry postings list:
- \[47\]

Work out how many comparisons would be done to intersect the two postings lists with the following two strategies. Briefly justify your answers:
- a. Using standard postings lists
- b. Using postings lists stored with skip pointers, with a skip length of $\sqrt{P}$, as suggested in Section 2.3.