Lecture 1 Exercises

A. Shawn Bandy

August 31st, 2013

1. Consider the following document collection:

   - **Doc 1:** breakthrough drug for schizophrenia
   - **Doc 2:** new schizophrenia drug
   - **Doc 3:** new approach to treatment of schizophrenia
   - **Doc 4:** new hope for schizophrenia patients

   Provide both the term-document matrix and inverted index for this document collection. What documents will be returned for the Boolean query `schizophrenia` AND `drug`? For the query `for` AND NOT(`drug` or `approach`)?

2. Suppose that x and y are the respective postings-list sizes for the terms `Romney` and `Obama`. Assuming n documents in the collection. What are the worst-case running times for returning the list of documents that satisfy `Romney` AND NOT `Obama`? `Romney` OR NOT `Obama`? Explain.

3. Convert

   ```
   (a or b) and not(c or d)
   ```

   to disjunctive normal form. If a, b, c, and d are terms with respective postings-list sizes x, y, z, and w, compare the worst-case running times of the original query with the DNF query. Assume n documents in the entire collection.

4. Give an example where a,b, and c are terms with respective postings-list sizes $x < y < z$, and for which

   ```
   (a and b) and c
   ```

   will have a longer running time than

   ```
   (c and b) and a
   ```

5. Use the Integral Theorem to obtain the order of growth of the sequence

   $$1 + 1/\sqrt{2} + 1/\sqrt{3} + \cdots .$$

   $\sum_{i=1}^{m} f(i) = \theta(integral f(x) from 1 to m dx)$

6. Repeat Example 5, but now use $c = 0.5$, followed by $c = 2$. Compare the memory usage for both cases with the $c = 1$ case computed in Example 5.

7. Consider a collection of n documents and m terms, where, for each document d and term t, t appears at most once in d. Moreover, if $t_1, \ldots, t_m$ is a listing of terms in decreasing order of frequency, assume that t_i occurs in $1/i$ of the documents. a) Give a big-Θ estimate of the number of tokens contained in this collection. b) What percentage of the most frequently occurring terms account for 80% of these tokens? Note: Pareto's rule would imply an answer of 20%. Is this correct?

T = of 1's in term_doc matrix

T = n + n/2 + n/3 ... n/m

m is arbitrary

m = sqrt(T) = sqrt(n log m) what does n have to be to make this an actual equality?

8. For the document collection of the previous problem, how would n have to depend on m in order for Heap's Law to be obeyed. In other words, what should we replace n by in terms of m so that the square root of the number of tokens is on the order of m?

9. Try the following Google queries: burglar, burglar burglar, and burglar OR burglar. Look at the estimated number of results and top hits. Are they identical (as they should be)?

10. Try the Google queries knight, conquer, and knight OR conquer. Are the number of hits for the third query bounded by the sum of the number of hits for the first two queries (as they should be)?