Information Retrieval	CS60092
Spring Semester, 2016	Mid-Sem
Maximum Marks: 50	Time Limit: 2 Hours

This exam contains 2 pages (including this cover page) and 6 problems.

You may *not* use your books and notes for this exam. Be *precise* in your answers. All the *sub-parts* of a problem should be answered at *one place* only. On multiple attempts, *cross* any attempt that you do not want to be graded for.

There are no clarifications. In case of doubt, you can take a valid assumption, state that properly and continue.

1. (7 points) Suppose you want to retrieve documents for the wildcard query "*ote*"? Explain how would you use

   (a) (4 points) Permuterm index

   (b) (3 points) Bigram index

   for the retrieval process. Take example of term "hotel" to explain. [Be very precise.]

2. (5 points) For a given corpus, suppose you plot $log_{10}T$ on $x - axis$ and $log_{10}M$ on $y - axis$, where T and M correspond to the number of terms in vocabulary, and the collection size, respectively. Suppose that the straight line with the best least square fit has a slope of 0.52, and intercepts the $y - axis$ at 1.5

   (a) (3 points) Use this information to compute the constants K and b for the Heaps' Law.

   (b) (2 points) The above corpus included the raw words. Suppose you do spelling correction as well as stemming. What would be the effect on these constants? [Only mention: which parameters will increase/decrease/remain constant]

3. (10 points) Suppose you want to index corpus from a new language, for which average word size (number of characters) per token is 8, and the average word size per term is 12. Also, it is not very uncommon to have words having 24 characters in this language, so you assign 24 bytes per term for the dictionary storage.

   (a) (2 points) Assume that in your corpus, you have 3 million unique words. Estimate the size of dictionary, while using the standard array of fixed width entries.

   (b) (4 points) How much compression can you achieve on this, if you store dictionary as a (long) string, with pointer to the next word showing end of the current word? [Report the final size of the dictionary]

   (c) (4 points) On top of that, suppose you use blocking with 8 strings in a block? What would be the additional saving? [Report the size after this step]

4. (10 points) Consider 4 documents in a collection, along with the terms in these documents, as shown in the figure below.

Doc 1: whale, sea, sea, whale, boat, boat, boat, boat, boat
Doc 2: whales, sea, sea, water
Doc 3: whale, water, water, whale, whale
Doc 4: whales, whales, whales

(a) (6 points) Suppose the collection contains 1000 documents in total. Given a user query, "whale boat water", use the lnc.lnc scheme to assign relevance scores to these 4 documents with respect to the query. No stemming. [Give relevance scores for these 4 documents]

(b) (4 points) Suppose your system shows top 3 documents as a result (as per relevance computed in part a), and the user marks 2nd document as relevant. If you apply relevance feedback (with $\alpha = 1, \beta = 1, \gamma = 0$) and rerank the documents, what would be the new ranking? [Show the new randking in decreasing order]

5. (8 points) The figure below shows the output of an information retrieval system on two queries. Crosses correspond to the relevant documents, dashes to non-relevant documets. Let the two documents contain 5 and 10 relevant documents, respectively, but only those shown in the figure are retrieved by the system, not the others.

(a) (4 points) Compute the mean average precision (MAP) of the system.

(b) (4 points) Compute the NDCG of the system at rank 5. Assume that each document has the same relevance.

Rank	Q1	Q2
1	X	–
2	–	X
3	–	–
4	X	X
5	X	–
6	–	X
7	–	X
8	–	–
9	–	X
10	–	X

6. (10 points) Answer the following questions:

(a) (3 points) For efficient processing, in what order should the following boolean query be processed?
(friends OR romans) AND (skies OR antony) AND (trees OR cleopatra)
The frequencies of various terms are given as follows:
friends: 25176, romans: 4582, skies: 34000, antony: 460, trees: 18000, cleopatra: 390

(b) (3 points) Write the gamma codes for the numbers 18 and 33.

(c) (4 points) In the Binary independence model, the retrieval status value (RSV) finally boils down to $\sum_{x_i=q_i=1} c_i$
where $c_i = log\frac{p_i(1-r_i)}{r_i(1-p_i)}$, $p_i = p(x_i = 1|R = 1, q)$ and $r_i = p(x_i = 1|R = 0, q)$
Explain how this can be approximated using only the parameters, obtained from the corpus?