Algorithms in the Real World
Assignment #4 Due Date: 4 December 02

Problem 1: (10pt) Prove the Eckart and Young theorem mentioned in the last paragraph on page 348 of the paper "Matrices, Vector Spaces and Information Retrieval". Note that this should be a relatively short proof and easy to generate from just the results in that paper. How might one use the theorem to help select k?

Problem 2: (20pt) Consider a term-document matrix A, and assume the columns are all normalized. If we consider the matrix A^TA, it gives a similarity matrix on documents based on the cosine measure. Consider the undirected weighted-graph corresponding to this matrix (it is undirected since the matrix is symmetric). Now consider running a graph partitioning algorithm that recursively divides the graph in two equal sized parts by trying to minimize the sum of the edge-weights that are cut. This generates a separator tree. Explain how you might use this tree to better compress the inverted index of A.

Problem 3: (20pt) In the discussion of merging with Treaps (see lecture 1 slides) we introduced the random variables A_{ij} and C_{ilm} along with their expectations a_{ij} and c_{ilm}. To bound the expected pathlength from the start to the l^{th} element we analyzed the term $\sum_{i=1}^{l} a_{i1}$ but not the term $\sum_{i=1}^{n} (a_{il} - c_{i1l})$. Please analyze this sum by writing an expression of c_{ilm} and taking the sum.

Problem 4: (20pt) Suppose that Bob has two n-bit secrets that Alice wants to learn. Alice is allowed to ask Bob questions about his secrets, where each question has a 1-bit answer and is applied to a single secret. Bob chooses which secret to apply each question to before hearing the question, and then answers the question truthfully. For example, Bob might have two secrets, 010 and 110, and might start by selecting his first secret, 010. If Alice then asks Bob, "Does your secret begin with a 0?", Bob would answer yes. Suppose that, over time, Bob chooses both of his secrets infinitely often. Show that if Alice has a deterministic algorithm for asking questions, it may not be possible to deduce either of Bob's secrets. (There is some subtlety to this argument.)

Problem 5: (10pt) Design a randomized algorithm for Alice that will eventually learn both of Bob's secrets.

Problem 6: (20pt) Suppose that, for each question, Bob now chooses which of the two secrets to apply the question to with probability 1/2, independent of any past random decisions. Design a deterministic algorithm for Alice that will learn both n-bit secrets with high probability (i.e., probability at least $1 - 1/n$), and prove an upper bound on the total number of questions that Alice asks.