TERMS OF ENDEARMENT

#a IDF: terms that appear in many documents are less likely to be important than terms that appear in few documents, because the more documents a term appears in the more likely the content of those documents are to be unrelated to the term. Therefore, greater inverse document frequency indicates greater importance.
TF: a term that appears many times in a document is an indicator that the term is important to that document and therefore that the document is more likely to be "about" that term. Therefore, greater term frequency indicates greater importance.

#b Stopword removal: By eliminating the terms with very long inverted lists.
Stemming: By reducing the number of inverted lists by consolidating the lists for two or more terms with the same stem.

TOTAL RECALL

#a There are 10 documents in the result set. 10 = (0.25 * 16) / 0.4.

#b Engine 1 ranks more relevant documents in the top results, but Engine 2 finds all of the relevant documents faster. If there are *R* relevant documents total, and precision at recall=1 is roughly 0.25 for Engine 2, then I have to go to rank *R*/0.25 = 4*R* in Engine 2 to find all of the relevant documents. If precision at recall=1 is roughly 0.01 for Engine 1, then I have to go to rank *R*/0.01 = 100*R* in Engine 1 to find all of the relevant documents. That's 25 times more documents I have to look at in Engine 1's results. Thus, I prefer Engine 2.

THE HOBBIT

#a Heaps' law tells us that $10000 = 10 \times (5000 \times 200)^{0.5}$. If we add 2200 documents, we have $V = 10 \times (7200 \times 200)^{0.5} = 12000$ vocabulary terms, so the bit vector index requires $7200 \times 12000 \approx 9 \times 10^7$ bits.

#b See IR textbook, Table 5.5. 24 = 11110,1000. Should be clear how the student arrives at the result.

NO STRINGS ATTACHED

#a See Aho-Corasick paper and course slides. Efficiently search for many strings simultaneously within a text, in a way that scales well with dictionary size: Organize the dictionary of words to search for in a trie, do a "trie walk" scanning through the text once. For NLP applications we might want to consider only token boundaries as valid start/stop locations.

#b See IR textbook, section 3.3.3. See also Shang-Merrett paper and course slides. Edit distance is defined as the smallest number of edit operations needed to rewrite one string to the other. An edit operation is a substitution, deletion or replacement – possibly also a transposition. Should very briefly describe the layout of the edit table *T*. Fill *T* starting at top

left corner, answer (edit distance) is at the bottom right corner. Filling happens such that *T(i,j)* is a function of values to the W, N and NW (and possibly also NWNW if we include transpositions) of *T(i,j)*.

#c Exploit that two strings in *D* that share a prefix of length *n* will also have identical first *n* columns in the table *T* – so organize the strings in *D* in a trie and traverse this trie, updating only the relevant column corresponding to the current trie depth. We can prune the search space and abort the search in a complete sub-trie as soon as the edit distance exceeds *k*. We can also exploit how column values in *T* start to increase after having reached a minimum and not have to update the complete relevant column, either ("Ukkonen's cutoff").

MY NEIGHBOR TOTORO

#a,b,c See IR textbook, chapter 14.

THE POSTMAN ALWAYS RINGS TWICE

#a See IR textbook, section 7.1.5. Should discuss what they are, pros/cons, when they are and aren't possible to use. Some keywords: Concurrent traversal, memory usage for accumulators, impact ordering, ordering by *g(d)*, etc.

#b Yes, e.g., using skip lists. See IR textbook, section 2.3. Should discuss what they are, pros/cons, when they are and aren't possible to use. Some keywords: Create at indexing time, intersections and not unions, distribution of skips, when they might have an adverse effect.