# Exam 1 review

I will let you bring one 3-inch by 5-inch index card of notes to the exam. You may write or print on both sides of the card.

## Basics

* Boolean retrieval system basics: query operators, the Boolean retrieval model, processing Boolean queries
  + do intersection/merge routines for two postings lists by hand for AND, OR, and AND NOT operators
  + know big-Oh complexity of the merge routines in terms of the lengths of the two lists being merged
* Vocabulary: corpus, document, term, information need, query, relevance, precision, recall, vocabulary, dictionary, index, posting, postings list, document frequency, term frequency, position
* Book: Chapters 1-5. Ignore 2.2.2, 2.3, "Permuterm indexes" in 3.2.1, 3.4, 4.2, 4.4, 5.1, 5.2.2, ignore the heavy math in 5.3

## Indexes

* Term-document matrices: construct a matrix, use a matrix to answer a question, calculate memory requirements for a matrix of a particular size
* Inverted index: use an inverted index, construct an inverted index by hand, compare and contrast with a term-document matrix
  + Positional inverted indexes and their role in phrase queries
  + Biword indexes and their shortcomings
  + Data structures for in-memory inverted indexes
* K-gram indexes: construct k-grams for a vocabulary type, and for a wildcard query; build a small k-gram index for a few types and use it to answer a wildcard query; the role of the post-filtering step in wildcard queries

## Vocabulary and terms

* Token vs. type vs. term.
* Issues in parsing documents (encoding types, language detection, irrelevant information, etc.)
* Tradeoffs in document unit sizes and indexing granularity choices (Chapter 2).
* Issues in tokenizing a document (hyphenation, non-alphanumeric characters, compound words, accents/diacritics, names; do not simply study the requirements for project 1, understand there are many more issues and solutions than what we have chosen)
* Spelling correction: schemes for determining that spelling correction should take place; Edit Distance algorithm; using k-grams for spelling correction; Jaccard coefficients
* Wildcard queries with k-gram indexes
* Using binary search trees for leading or trailing wildcards (Chapter 3)

## Large Scale Search Engines

* Constructing an index with SPIMI
* Dynamic indexing: strategies for dealing with a changing corpus. Logarithmic merging, rebuild-from-scratch. (Chapter 4.)
* Index-on-a-disk: know the complete picture of a search engine that keeps all information (index, vocabulary) on disk.
  + be able to construct an on-disk representation of an index from an existing corpus
  + be able to read postings from an on-disk representation of an index
* Encoding gaps vs. encoding document ids. (Chapter 5.)
* Compression: variable byte codes. Compute codes for various numbers.
* B+ trees for vocabulary retrieval
  + calculate optimal B+ tree factors (M and L) given information about the average key size, the value size, and the block size
  + do not need to insert or remove items from a tree
* Why indexes need to be on disk, what the major performance factors are for disk-based systems.
* Hardware issues: block sizes, disk seek times, caching. (You do not need to know numbers for any of these things, just what they refer to, and how they impact the system as a whole.)

## Algorithms to Know

* Postings list merges
* SPIMI
* Edit distance
* Jaccard coefficient calculation
* Variable byte encoding and decoding
* Logarithmic merging