CS 363 Outline for Midterm Exam

Use appendix A and exponent formulas.

Algorithm analysis
- Worst case analysis
- Average case analysis in simple, equally likely situations
- Idea of lower bounds applying to all algorithms in a class
- Lower bounds via decision trees

Growth order of functions:
- sequence of orders for common functions
- simplifying order expressions
- big and small Oh, theta, and omega

searching
- linear, binary

Recurrence relations:
- Master theorem! Be able to classify recurrence relations and use it!
- Creating recurrence relations T(n) = f(n) + terms involving T
- Distinguish non-recursive term f(n) from recursive terms and from T(n)!
- Formulas for T(n) as sum if T(n) = T(n-1) + f(n)

- Strassen's algorithm as example

Sorting. For all: follow algorithm for small data sets, stable sorts
- Insertion sort algorithm, worst, best, average cases
- Selection Sort
- Quicksort,
  - initial partition algorithm,
  - stack issues, version with loop
  - worst case, best case, order of average case
- MergeSort
  - Analysis for n = power of 2
  - Lower bound for key comparison based search via decision tree
- Heapsort
  - Best, worst case behavior

Radix sort
- Order analysis

Counting Sort
- Order analysis

Comparisons of all sorts (where each has advantages)

Idea of amortizing
- Array copying for stack as example of amortizing

B-trees
- Algorithms for search, insert, delete – apply to specific cases
- Order of algorithms
- Relation to red-black trees

Hash tables - algorithms for chained and open (linear and double);
- memory usage; average, worst case behavior