CISC 621 Algorithms, Midterm exam guide
(Exam is in class, March 24)

The midterm exam will have 5 parts.

Part 1: Some multiple choice and short answer questions asking about basic facts.

For an example:
$\log^*(x) = 0$, if $x = 1$.
$\log^*(x) = 1 + \log^*(\lg(x)$, if $x > 1$.
Which is true?

(a) log* (16) = 3.
(b) There are many more integers n for which log* (n) = 5 than there are for which log* (n) = 4.
(c) If log* (n) = 6, then n is a lot larger than the gross domestic product of the USA (in dollars).
(d) all of the above.

Part 2: [a correctness argument] Select one of the following two algorithms and explain why it correctly solves the corresponding algorithmic problem P, [a choice among 2 would be offered. See the indices of problems and algorithms in the course notes for the overall list from which the choices will be taken.] Each of the two algorithms and corresponding problems would be explicitly stated on the exam paper - you then explain why it works.]

Part 3: [a runtime analysis] Select one of the following two algorithms. State and explain the worst case runtime of the algorithm on inputs of size n. [list of two algorithms]

Part 4: [Data structure explanation] We have studied priority queues (binary heap and binomial heap implementations so far), binary search trees, 2-3-4 trees, red-black trees, and dynamic disjoint sets (union-find). Be prepared to explain about structure and costs of one of these.

Part 5: Another question like one of parts 2, 3, or 4.

Explanations will be graded for clarity and thoroughness as well as correctness.

List of Problems and algorithms

For each algorithm or data structure implementation below you should know how it works, the costs (worst case, amortized, or expected, as appropriate). For *how it works*, know how to prove(explain) correctness. For *the costs*, know how to give the analysis, with emphasis on the upper bound (big-O part).

- Problem Max-min(A,n): Find both the maximum and minimum of an (unordered) array of n numbers. Algorithm in about 3n/2 comparisons, lower bound proof.
- Problem median(A,n): Find the element in an (unordered) array of n numbers that would be in the $\lfloor n/2 \rfloor$ position if the array were sorted. Algorithm: use select.
- Problem select(A,n,k): Find the element of rank k in unordered array A of length n. [side issue: know meaning of *rank* in this context.] Algorithms: expected linear time `Randomized-Select` and worst case linear time `Select`,
- Problem sort: Algorithms insertionSort, heapSort, mergeSort, quickSort, introspectiveSort [Ch 2,6,7]
- Problem polynomial multiplication. Algorithm Karatsuba's divide and conquer.
- Data Structure (Min)-Priority Queue (insert, extractMin). Implementation: binary heap, binomial heap. [Ch 6, exercise in Ch 19]
- Data Structure Dictionary (insert, delete, search(=find), first, last, prev, next) Implementation: Left Leaning Red-Black trees [Ch 13, Sedgewick slides]

Tools: Master theorem [Ch 4], randomization [Ch 5]