COMP 3760 – Final exam review guide

Makeup of the exam:

- Generally similar to the midterm
- ~75% T/F and M/C
- ~25% Long answer (describing an algorithm to solve a problem, writing pseudocode, etc.)
- Cumulative but heavily weighted (80-90%) to the material from Weeks 7-14

Study suggestions:

- Review all course slides
- Pay attention to all mentioned terminology, definitions, etc.
- Make sure you understand how each algorithm works in general terms
- Trace through examples by following the pseudocode

Get a good night's sleep on Sunday night!

Week-by-week overview:

Weeks 1,2 – Analysis of algorithms, efficiency, orders of growth

Determine the basic operation of some pseudocode

Set up a summation formula to represent the efficiency of some pseudocode

Determine the efficiency/running time of a pseudocode algorithm

Week 3 – Brute force algorithms

Sorting algorithms - selection, bubble - know how to perform them, know the efficiency

String matching problem - know the brute force approach (how to perform)

Traveling salesman problem - know the brute force approach, know the efficiency

Knapsack & assignment problems - know the problems, know the efficiency of brute force approaches

Week 4 – Decrease and conquer algorithms

Three types (decrease by constant, decrease by constant factor, decrease by variable amount)

Insertion sort - know how to perform, know efficiency

Generating permutations and subsets - how to perform

Binary search - how to perform, know efficiency

Fake coin problem - how to perform

Week 5 – Divide and conquer algorithms

Know the general principles, how to distinguish from "decrease and conquer"

Mergesort - know how to perform, know the efficiency

Binary tree problems - know the typical recursive pattern, know the efficiency

Week 6 – Transform and conquer algorithms

Know the general principles (instance simplification, representation change)

Pre-sorting - overall advantage, when/why it's useful

Heapsort - know the basic algorithm, know whether a heap is valid, convert between array/tree representations, know the efficiency

Week 7 – Space/time trade-offs

Two varieties: "input enhancement", "pre-structuring"

Comparison counting sort - describe the basic principle, be able to trace the pseudocode, state the efficiency

Distribution counting sort - describe the basic principle, be able to trace the pseudocode, state the efficiency

String matching - Horspool's algorithm - describe the basic principle, perform the algorithm

Hashing - describe the basic principle, describe the two strategies for collision handling, perform the algorithms

Week 8 – Data structures and graphs

Know the principles of all the common data structures

Graphs - know both adjacency list and adjacency matrix representations

DFS and BFS - describe the basic principle, be able to trace the pseudocode, state the efficiency

Weeks 9 – Graph algorithms

Topological sort - two algorithms (DFS, "decrease by one") - describe the basic principle, be able to perform both

Binary tree traversal - Inorder, Preorder, Postorder - describe all three, calculate them for a given tree, given two calculate the other

Week 10 – Greedy algorithms

Know the general principles

Change-making problem - understand the problem, know the greedy approach, its limitations

Minimum spanning trees - Prim's and Kruskal's algorithms - know the algorithms, know the differences, be able to perform both, recognize problems that this would apply to

Disjoint subset/union-find - know the principle, be able to perform, know why it's used in Kruskal's algorithm

Single-source shortest path - Dijkstra's algorithm - know the algorithm, be able to perform, recognize problems that this would apply to

Graph coloring - know the principle, know how it relates to map coloring, be able to apply the greedy algorithm to the problem

Week 11 – Dynamic programming

Know the basic principle (define problem recursively, save/recall results of subproblems), be able to apply to an appropriate problem

Warshall's algorithm - transitive closure, know how to apply to a graph

Floyd's algorithm - all pairs shortest paths, know how to apply to a graph

Week 12 – Backtracking, branch and bound, decidability

Be able to describe the state space tree for an applicable problem

Backtracking - know the general principle, how to apply to n-queens problem

Branch and bound - know the general principle, key difference with backtracking, how to apply to an applicable problem

Decidability - just remember the utter futility of it all ☺