Math 260
H. Jordon

Study Guide for Exam 2

In what follows is a **brief** synopsis of what we have covered in Sections 4.1–4.5, 5.1–5.3. Use this list a guide to help you make up your own study guide.

On the exam, you can expect several proofs, TRUE/FALSE questions, and give-an-example-of type questions. The problems that have been assigned (but not necessarily graded) or very similar problems could appear on the exam; therefore it is **highly** recommended that you make every effort to understand **ALL** problems assigned.

Exam 2 Topics:

1. Graphs, adjacent, incident, degree, subgraph.
2. Complete graphs K_n and complete bipartite graphs $K_{m,n}$.
3. First Theorem of Graph Theory (Theorem 4.1) and its Corollary.
4. Adjacency matrices.
5. Isomorphism, showing two graphs are isomorphic, showing two graphs are not isomorphic, graph isomorphism invariants.
6. Multigraphs, parallel edges, loops.
7. Paths, simple paths, circuit, cycles.
8. Connected graphs, minimum number of edges in a connected graph with n vertices.
9. Euler paths and euler circuits; necessary and sufficient conditions for a multigraph to have an euler circuit or euler path.
10. Hamiltonian cycles and paths; Theorem 4.6 – a sufficient condition for a graph to have a hamiltonian cycle (why is this not a necessary condition for a graph to have a hamiltonian cycle?).
11. Shortest paths and distance; Dijkstra's Algorithm for finding the shortest path in a weighted graph.
12. Coloring the vertices of a graph; chromatic number of a graph; Greedy Coloring Algorithm.
13. Directed graphs and multigraphs, indegree, outdegree, adjacency matrices, First Theorem of Directed Graphs (Theorem 4.10).
14. Directed paths and cycles, directed euler paths, directed euler cycles.
15. Tournaments.

16. Trees, properties of trees (Theorems 5.1, 5.2, 5.3, 5.4).
17. Equivalent conditions for a graph to be a tree (Theorem 5.5).
18. Prufer codes: constructing the code from the tree and the tree from the code.
19. Spanning trees.
20. Minimum cost spanning trees, Prim's and Kruskal's Algorithms.
21. Searching in a tree, Breadth-first searching and Depth-first searching.
22. Using depth-frst searching and back-tracking to find a solution to a problem, like the 4-queens problem.