In order to receive full credit problems and solutions must be clearly written and presented neatly. In writing up your solution, you should state the problem as explaining the solution. Merely giving answers to exercises will is not enough to receive credit. Your write up must show work and convince me that you understand the material. Late homework will not be graded. Starred (★) proof problems are eligible for inclusion in the proof portfolio assignment.

- Choose, complete, and write-up **3 exercises** and **3 proofs**.
- Exercises are worth up to 3 points, and proofs are worth up to 5 points.
- Put your name on each page you submit, and somewhere indicate who you worked with on this assignment.

Exercise 8.1: From the text, Exercise 3.4.

Exercise 8.2: From the text, Exercise 3.6.

Exercise 8.3: From the text, Exercise 3.8.

Exercise 8.4: Find and draw all non-isomorphic bipartite simple graphs with exactly 5 vertices and at least one cycle.

Exercise 8.5: Find a reference for Ford's algorithm for minimal paths and explain it. Using an example demonstrated how Ford's algorithm works.

Exercise 8.6: For the following degree sequences (list of the degrees of the vertices in a graph) determine if there exists a bipartite simple graph with these degrees. If yes construct it, if not explain why it does not exist. (a) (6, 6, 4, 4, 4, 4, 4, 4, 2, 2, 2); (b) (4, 4, 4, 4, 4, 3, 3, 2, 2); (c) (5, 5, 5, 5, 5, 5, 4, 4, 4)

Proof 8.1: Prove that if a simple graph contains an odd length closed trail $v_0 \to v_1 \to v_2 \to \cdots \to v_k$ then the graph contains an odd length cycle.

Proof 8.2: Prove that if each component of a graph is bipartite, then the entire graph is bipartite.

Proof 8.3: From the text, Exercise 3.12 b) (complete matching is defined in 3.11).

Proof 8.4: ★ From the text, Exercise 3.16. Hint: You must prove by induction that the number of spanning trees h_n is equal to F_{2n-1}. Do this by proving a recurrence relation for h_n.

Proof 8.5: ★ Prove that a simple graph with 17 vertices and 73 edges cannot be bipartite. (Hint: Break the graph up into two vertex sets and count the number of edges within each vertex set.)