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 4.1: From the text, Exercise 1.14.

Exercise 4.2: From the text, Exercise 1.13.

Exercise 4.3: From the text, Exercise 1.16.

Exercise 4.4: Consider the sequence of numbers $u(1), u(2), u(3)$, ... where $u(n+1) = [u(n)^2 + 4]/[u(n)+2]$ As $n \to \infty$, $u(n) \to a$ Find a, and show that it's the limit.

Exercise 4.5: Let $q_0 = \alpha$, $q_1 = \beta$ and $q_n = (1 + q_{n-1})/q_{n-2}$. Assuming that $q_n \neq 0$ for all n, solve this recurrence relation. Hint: $q_4 = (1+\alpha)/\beta$

Exercise 4.6: Solve the recurrence $x_n = n$ for $0 \leq n < m$ and $x_n = x_{n-m} + 1$ for $n \geq m$.

Exercise 4.7: Each day Angela eats lunch at a deli, ordering one of the following: chicken salad, a tuna sandwich, or a turkey wrap. Find a recurrence relation for the number of ways for her to order lunch for the "n" days if she never orders chicken salad three days in a row.

Exercise 4.8: What is dynamic programing? Describe the basic algorithm and explain it's connection to recurrence relations.

Exercise 4.9: What is the Josephus number? Define it and give it's historical origins. What is it's connection to recurrence relations.

Exercise 4.10: A string that contains only 0s,1s, and 2s is called a **ternary string**. Find a recurrence relation for the number of ternary strings that contain two consecutive 0s? What are the initial conditions? Explain your answer.

Proof 4.1: Prove, using induction, that for the sequence defined by $a_1 = 0$, $a_2 = 1/2$, and $a_{k+2} = \frac{1}{2}(a_k + a_{k+1})$ for $k \geq 1$ that

$$a_n = \frac{1}{3}\left(1 - \left(-\frac{1}{2}\right)^{n-1}\right)$$

for every $n \geq 1$.

Proof 4.2: Consider the Fibonacci sequence defined in the standard way $f_1 = 1$, $f_2 = 1$, and $f_n = f_{n-1} + f_{n-2}$. Prove that

$$\sum_{i=1}^{3m+1} f_i$$

is always odd for m an non negative integer.

Proof 4.3: ★ From the text, Exercise 1.11.

Proof 4.4: From the text, Exercise 1.12.

Proof 4.5: ★ From the text, Exercise 1.17.

Proof 4.6: ★ From the text, Exercise 1.18.

Proof 4.7: ★ Let $F_{n+2} = F_{n+1} + F_n$ with $F_0 = 0$ and $F_1 = 1$, the Fibbonacci numbers, and $L_{n+2} = L_{n+1} + L_n$ with $L_0 = 2$ and $L_1 = 1$, the Lucas numbers. Prove

$$\sum_{k=0}^{n} 2^k L_k = 2^{n+1} F_{n+1}$$

for all $n \geq 0$.

Proof 4.8: ★ Let $F_{n+2} = F_{n+1} + F_n$ with $F_0 = 0$ and $F_1 = 1$, the Fibbonacci numbers, and $L_{n+2} = L_{n+1} + L_n$ with $L_0 = 2$ and $L_1 = 1$, the Lucas numbers. Prove

$$\sum_{k=0}^{n} (-1)^k 2^{n-k} L_{k+1} = (-1)^n F_{n+1}$$

for all $n \geq 0$.